

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

CONTROLADOR BASEADO EM LÓGICA NEBULOSA APLICADO A UM
SISTEMA SERVOPNEUMÁTICO
(SIMULAÇÃO)

DISSERTAÇÃO SUBMETIDA A UNIVERSIDADE FEDERAL DE SANTA
CATARINA COMO REQUISITO PARCIAL PARA A OBTENÇÃO DO TÍTULO
DE MESTRE EM ENGENHARIA.

Marco Antonio Silva Picanço

Florianópolis, julho de 1999

CONTROLADOR BASEADO EM LÓGICA NEBULOSA APLICADO A
UM SISTEMA SERVOPNEUMÁTICO.


(SIMULAÇÃO)

Marco Antonio Silva Picanço

DISSERTAÇÃO APROVADA COMO REQUISITO PARCIAL PARA A
OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA, ESPECIALIDADE
ENGENHARIA MECÂNICA E APROVADA EM SUA FORMA FINAL PELO
CURSO DE PÓS-GRADUAÇÃO

Prof. Carlos Alberto Martin Dr-Ing. 

ORIENTADOR

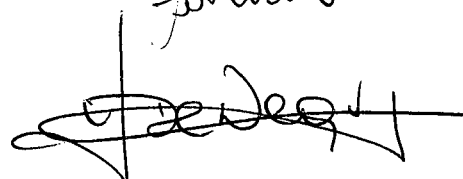
Prof. Dr. Júlio César Passos 

COORDENADOR DO CURSO

BANCA EXAMINADORA:

Prof. Dr. Guilherme Bittencourt 

Prof. Dr. Antonio Augusto Rodrigues Coelho 

Prof. Dr. Victor Juliano De Negri 

Florianópolis, 01 de julho de 1999.

Aos meus pais.

A minha avó. (In memorium)

AGRADECIMENTOS:

Agradeço como discípulo ao Prof. Czeslau Barkzak que me iniciou nos estudos da inteligência artificial; ao Prof. Carlos Alberto Martin pelo acolhimento e orientação; a Cristiane e Jackson Azevedo e aos meus amigos Eduardo Alves `Dudu` Portela Santos, Etney `El Magron` Neves e Fred Amorim pela ajuda mútua nas horas difíceis e nas alegrias em terras catarinenses; a Alessandra pela compreensão e carinho; a agência financiadora CNPq que presta um importante papel na soberania intelectual e científica deste país.

SUMÁRIO

SÍMBOLOS E ABREVIATURAS	xi
INTRODUÇÃO	1
CAPÍTULO 1	7
1 MODELAGEM DE SISTEMAS SERVOPNEUMÁTICOS.....	7
1.1 CARACTERIZAÇÃO DE UM SISTEMA SERVOPNEUMÁTICO	7
1.2 ANÁLISE NÃO-LINEAR DO SISTEMA SERVOPENUMÁTICO	9
1.3 ANÁLISE LINEARIZADA DO SISTEMA SERVOPNEUMÁTICO	16
CAPITULO 2	38
2 A LÓGICA NEBULOSA	38
2.1. CONTROLADOR BASEADO EM LÓGICA NEBULOSA (CLN)	42
2.2 CONSIDERAÇÕES	63
CAPITULO 3	64
IMPLEMENTAÇÃO E RESULTADOS	64
3.1 MODELO DO SERVO-POSICIONADOR PNEUMÁTICO	64
3.1.2 FUNÇÃO TRANSFERÊNCIA:	66
3.2 CURVAS EM MALHA ABERTA	68
3.3 ALGORÍTMOS DO CONTROLADOR FUZZY	70
3.4 CONTROLADORES E CURVAS EM MALHA FECHADA	79
3.5 AVALIAÇÃO DO DESEMPENHO	105
CAPÍTULO 4	109
CONCLUSÃO.....	109
REFERÊNCIAS BIBLIOGRÁFICAS	111
APENDICE	118
LISTAGENS	118
FDS.....	118
FLUXOGRAMA FDS	122
FDSV.....	123

FLUXOGRAMA FDSV	125
FDL	126
FLUXOGRAMA FDL.....	129
FMR	130
FLUXOGRAMA FMR.....	133
FLC	136
fuzzyfy.c	139

FIGURAS

Figura 1- Variáveis de um sistema servo pneumático.....	10
Figura 2- Coeficiente Ψ para correção da vazão mássica.....	12
Figura 3 – Exemplo de um diagrama de um sistema pneumático usual.....	18
Figura 4 – Representação de um sistema massa+mola clássico e um sistema servopneumático.	23
Figura 5 – Identificação do volume de controle e da superfície de controle....	30
Figura 6 – Sistema servopneumático e sua função de transferência linear de terceira ordem.	33
Figura 7 - Representação do processo de um controlador baseado em Lógica Nebulosa.	42
Figura 8 –Representação de um Conjunto Nebuloso e seus pontos significativos.	45
Figura 9 – Formas comuns para funções de pertinência.	46
Figura 10 – Mapa de uma Variável Linguística em um processo de Nebulização.....	47
Figura 11 – Exemplo de entrada em um processo de Nebulização.....	48
Figura 12 – Formato de uma regra constante de um bloco de avaliação de regras.	50
Figura 13 – Exemplo de tabela verdade para três variáveis lingüísticas.	51
Figura 14 –Conjuntos Nedulosos	53
Figura 15 – Operação e/\cap sobre os conjuntos nebulosos da Figura(14).....	53
Figura 16 – Operação ou/\cup sobre os conjuntos nebulosos da Figura(14).	54
Figura 17 – Operação Complemento/negação sobre os conjuntos nebulosos da Figura (14).....	54

Figura 18 – Entrada para a variável ERR.	56
Figura 19 - Entrada para a variável VERR.....	56
Figura 20 – Regra a ser avaliada.....	56
Figura 21 – Representação gráfica da avaliação de uma regra.	57
Figura 22 – Avaliação de um grupo de regras.....	58
Figura 23 – Desnebulização utilizando o método do centróide.	60
Figura 24 – Desnebulização utilizando o método do máximo.	60
Figura 25 – Desnebulização utilizando o método do Centro de Gravidade Ponderado.....	61
Figura 26 – Diagrama de Blocos de uma modelagem em terceira ordem de um sistema servo pneumático controlado por um CLN.	66
Figura 27 - Curva em malha aberta do modelo linear de terceira ordem com 50% de abertura da válvula (simulação). (SANTOS,1996)	68
Figura 28 – Curva em malha aberta experimental e teórica do modelo não- linear com 50% de abertura. (linhas pontilhadas – modelo não-linear ; linha cheia – resultado experimental) (SANTOS,1996)	69
Figura 29 - Curva em malha aberta do modelo linear de terceira ordem com 20% de abertura da válvula (simulação). (SANTOS,1996)	69
Figura 30 - Curva em malha aberta experimental e teórica do modelo não-linear com 20% de abertura. (linhas pontilhadas – modelo não-linear ; linha cheia – resultado experimental) (SANTOS,1996).....	70
Figura 31 – Árvore ternária.....	75
Figura 32 – TEMPO x VALOR FUZZYFICADO.	77
Figura 33 –Espaço alocado em disco para os diferentes algoritmos de Nebulização.....	78

Figura 34 – Arquitetura do CLN-PD.....	82
Figura 35 – Partição das Variáveis Linguísticas.....	82
Figura 36 – Topologia da Superfície da Saída do CLN-PD.....	83
Figura 37 – Controlador Lógico Nebuloso – PD (Posição) MASSA: 3,3 kg	84
Figura 38 – Controlador Lógico Nebuloso – PD (Esforço de controle) MASSA: 3,3 kg	85
Figura 39 – Controlador Lógico Nebuloso – PD (Posição) MASSA: 0,5 kg	85
Figura 40 – Controlador Lógico Nebuloso – PD (Esforço de controle) MASSA: 0,5 kg	86
Figura 41 – PID clássico (Posição) MASSA: 3,3 kg.....	88
Figura 42 – PID clássico (Esforço de controle) MASSA: 3,3 kg	89
Figura 43 – PID clássico (Posição) MASSA: 0,5 kg.....	89
Figura 44 – PID clássico (Esforço de controle) MASSA: 0,5 kg	90
Figura 45 – Arquitetura do controlador PID+CLN supervisorio	91
Figura 46- Topologia da Saída do CLN supervisorio direto	92
Figura 47 – PID+CLN supervisorio direto (Posição) MASSA: 3,3 kg	93
Figura 48 – PID+CLN supervisorio direto (Esforço de controle) MASSA: 3,3 kg	93
Figura 49 – PID+CLN supervisorio direto (variação do ganho) MASSA: 3,3 kg	94
Figura 50 – PID+CLN supervisorio direto (Posição) MASSA: 0,5 kg	94
Figura 51 – PID+CLN supervisorio direto (Esforço de controle) MASSA: 0,5 kg	95
Figura 52 – PID+CLN supervisorio direto (variação do ganho) MASSA: 0,5 kg	95

Figura 53 – PID+CLN supervisório inverso (Posição) MASSA: 3,3 kg	97
Figura 54 – PID+CLN supervisório in (Posição) MASSA: 0,5 kg	98
Figura 57 – PID+CLN supervi verso (Esforço de controle) MASSA: 3,3 kg ...	97
Figura 55 – PID+CLN supervisório inverso (variação do ganho) MASSA: 3,3 kg	98
Figura 56 – PID+CLN supervisório inverso sório inverso (Esforço de controle) MASSA: 0,5 kg	99
Figura 58 – PID+CLN supervisório inverso (variação do ganho) MASSA: 0,5 kg	99
Figura 59 – Arquitetura do CLN-PI + PD clássico.....	100
Figura 60 - Topologia do CLN-PI .	102
Figura 61 – CLN-PI + PD clássico (Posição) MASSA: 3,3 kg	103
Figura 62 – PID+CLN supervisório inverso (Esforço de controle) MASSA: 3,3 kg.....	103
Figura 63 – CLN-PI + PD clássico (Posição) MASSA: 0,5 kg	104
Figura 64 – CLN-PI + PD clássico (Esforço de controle) MASSA: 0,5 kg ...	104

SÍMBOLOS E ABREVIATURAS

Π	-	Número de Mach
ξ	-	Amortecimento
γ	-	c_p/c_v
β	-	Módulo de Elasticidade Volumétrica
φ	-	Número Constante
λ	-	Variável auxiliar para o método de Lagrange
ω_n	-	Frequência natural
ω	-	Frequência natural da servoválvula
A_n	-	Área do embolo “n”
b	-	Razão crítica de pressão em escoamento compressível
c	-	Velocidade do som no meio
C	-	Coeficiente vazão/pressão de uma servo-válvula
c_{po}	-	Calor específico à pressão constante na estagnação
c_{vo}	-	Calor específico a volume constante na estagnação
f	-	Coeficiente linear de amortecimento e atrito viscoso
F_l	-	Força de carregamento
F_r	-	Força de atrito
h	-	Entalpia
h_o	-	Entalpia na estagnação
K_q	-	Ganho da servo-válvula (vazão/sinal de entrada)
M	-	Massa total acoplada ao sistema
m_n	-	Massa de ar na câmara “n”
\dot{m}_n	-	Variação temporal da massa de ar na câmara “n”

p_s	-	Pressão de suprimento (absoluta)
P_{at}	-	Pressão atmosférica
p_c	-	Pressão líquida no cilindro
p_n	-	Pressão na câmara “n” (absoluta)
\dot{p}_n	-	Variação temporal da pressão na câmara “n”
Q_c	-	Vazão Volumétrica líquida
Q_N	-	Vazão Normal
R	-	Constante Geral dos Gases
s	-	Entropia
S	-	Velocidade Linear do Fluido
t	-	Tempo
T	-	Temperatura
T_o	-	Temperatura na estagnação
u	-	Energia Interna
u	-	Sinal Elétrico (Capítulo 1)
v	-	Volume específico
V_n	-	Volume da câmara “n”
V_{no}	-	Volume Morto na câmara “n”
V_t	-	Volume Total
x	-	Deslocamento do cilindro
X_1	-	Deslocamento da extremidade da mola
X_2	-	Deslocamento da massa
X_e	-	Sinal elétrico de entrada na servo-válvula
y	-	Deslocamento do carretel da servo-válvula

RESUMO

Nesta dissertação são discutidos os aspectos pertinentes a utilização de controladores baseado em lógica Nebulosa em sistemas servopneumáticos. Um modelo matemático linear de 3ª ordem com perturbação de carga foi utilizado para representar o sistema servopneumático. Foram implementadas cinco estratégias de controle e realizada uma análise comparativa do seu desempenho. Por fim conclui-se que é factível a utilização de controladores baseado em lógica nebulosa para a solução de controle num sistema servopneumático.

ABSTRACT

In this work, some aspects of fuzzy logic based controllers when applied to a pneumatic servo system are discussed. A third-order linear model with load disturbance is used to represent the pneumatic system. Five control strategies, including a PID controller, are presented and evaluated. The feasibility of fuzzy logic controllers as control solution in pneumatic servo systems is concluded.

INTRODUÇÃO

A afirmativa de que com o advento de controladores inteligentes, as ferramentas matemáticas para análise, modelagem e controle de fenômenos físicos seriam abandonadas, é um exagero. Afirma-se que, em algumas aplicações onde existam dificuldades na identificação do sistema, severas não-linearidades e variações de parâmetros internos os controladores inteligentes justificam a sua vantagem sobre as demais técnicas de controle. Neste momento as ferramentas analíticas e matemáticas não seriam aplicadas e os algoritmos de inteligência artificial realizariam o papel de um controlador humano, que teria a sua experiência codificada numa base de conhecimento ou experimentaria o sistema e a partir do uso dos algoritmos aprendido formaria a melhor estratégia de controle.

Todavia, deve-se atentar para o fato de que, na maioria das aplicações industriais, as condições para realizar-se ensaios “*on-line*” não podem ser atendidas. Além disto, não são todos os controladores chamados inteligentes que possuem a sua capacidade de aprendizado implementada. Em algumas destas aplicações, características do comportamento dinâmico dos sistemas em foco permanecem obscuras, revelando-se, somente após a análise criteriosa da modelagem matemática e da simulação destes sistemas.

Nestas aplicações, o paradoxo entre a utilização de um controlador inteligente e a modelagem matemática de um sistema estaria resolvido e a necessidade de modelagem matemática destes sistemas dinâmicos justificada. A análise matemática do fenômeno físico, a avaliação da experiência e a capacidade cognitiva dos controladores inteligentes formam, em conjunto, a solução do controle destes sistemas.

Uma das tecnologias de controle baseadas em inteligência artificial é o controlador baseado em Lógica Nebulosa. Desenvolvida por LOFTI ZADEH a partir de 1965 (ZADEH, 1965) a Lógica Nebulosa tornou os métodos heurísticos e cognitivos acessíveis aos micro-controladores dedicados (MANDANI, 1974). Através de conceitos de vagueza e imprecisão a Lógica Nebulosa constitui as suas variáveis lingüísticas, conjuntos e algoritmos Nebulosos, possibilitando ao operador codificar a sua estratégia e protocolo de controle sobre um determinado sistema.

Tomando-se como base a sua capacidade cognitiva, a tecnologia dos controladores pode se dividir em quatro gerações. A primeira geração estaria representada pelos os controladores fundamentados na teoria de controle clássica, precursores de toda formulação de controle. Em seguida, pertencentes a segunda geração de controladores estariam os controladores baseados na teoria de controle moderno, multi-variável ou por realimentação de estados, que, igualmente aos controladores da primeira geração são dotados de pequena capacidade cognitiva. Num próximo estágio estariam os controladores adaptativos, formando a terceira geração de controladores, a

primeira a apresentar um maior grau de *inteligência*, trata-se de uma grande evolução na cadeia dos controladores para a sua utilização em sistemas não-lineares e complexos. Em contra partida, apresenta um grau elevado de complexidade matemática. Na quarta geração estariam os controladores baseados em inteligência artificial que possuem como finalidade a codificação da experiência do controlador humano. Controladores inteligentes já apresentam elevada capacidade cognitiva e aprendizado, dentro desta categoria encontram-se os controladores baseados em Lógica Nebulosa, Redes Neurais, Sistemas Especialistas e os seus híbridos. (FLYNN,1997)

São objetos desta dissertação os controladores baseados em Lógica Nebulosa, controladores pertencentes à quarta geração de controladores, cujas capacidades cognitivas serão desenvolvidas e explicitadas no decorrer deste trabalho de dissertação, visando a exploração de sua natureza não-linear para controle de um servoposicionador pneumático. No entanto o mesmo, não acontece com a sua capacidade de aprendizado que ficou fora do escopo deste trabalho.

Atualmente o estágio de desenvolvimento de controladores baseados em Lógica Nebulosa, não esta apenas vinculado à estrutura lógica formal constituída por ZADEH (1965), mas fundindo-se com as Redes Neurais, formando os seus híbridos e absorvendo modificações em seus parâmetros estruturais. Nos híbridos neuro-nebulosos são aliadas a capacidade de aprendizado das Redes Neurais com a robustez e convergência do controlador Nebuloso. Não apenas controladores estão sendo desenvolvidos com Lógica

Nebulosa, mas sistemas capazes de representar modelos não-lineares e complexos. (JANG, 1995 e GOMIDE e GUDWIN, 1994)

O objetivo deste trabalho é estabelecer uma perspectiva para utilização de controladores baseados em Lógica Nebulosa como possível solução de controle em sistemas servopneumático. Espera-se atingir este objetivo através da implementação, ajuste e simulação de um controlador baseado em Lógica Nebulosa e a avaliação dos seus resultados em condições de variação de carga. Sabendo-se que devido a sua baixa rigidez a dinâmica do sistema pneumático se mostra muito sensível a variações de carga. Serão abordados aspectos importantes para a implementação de um controlador Nebuloso, a avaliação do desempenho de diferentes algoritmos para a Nebulização e a sintonia de diferentes características de controle e seus respectivos resultados.

Foi escolhido um servo posicionador pneumático, devido à sua recente utilização em aplicações de servo posicionamento. Haja visto que, estudos sobre posicionadores servopneumáticos são tão recentes quanto os controladores baseados em lógica Nebulosa, originam-se da década de 50 no Massachusetts Institute of Technology (MIT). Além disto vale ressaltar que a falta de difusão sobre a aplicação de controladores baseados em Lógica Nebulosa atuando sobre plantas servopneumáticas, também foi um fator motivante para este trabalho.

Esta dissertação foi dividida em quatro partes fundamentais:

- *A modelagem matemática*
- *O controlador baseado em Lógica Nebulosa*
- *A implementação e os resultados obtidos*
- *Conclusões*

Na primeira parte é apresentada a modelagem adotada para o sistema servopneumático, que foi posteriormente utilizada para a implementação de uma simulação. Foram discutidas abordagens distintas entre si, a linear e a não-linear, onde a opção por uma modelagem linear é justificada.

Em seguida são detalhados os fundamentos de um Controlador Baseado em Lógica Nebulosa (CLN), enfocando seus aspectos principais, seus processos e a sua construção. Cada componente do Controlador Nebuloso é detalhado e são estabelecidas as suas inter-relações, utilizando uma abordagem clara, objetivando a sua implementação.

No espaço estabelecido para discussão sobre a implementação do controlador, apresentação e análise dos resultados, foram discutidos aspectos relevantes à implementação em linguagem de programação do processo de Nebulização do Controlador. Para a comparação do desempenho dos controladores foram estabelecidos parâmetros para análise. Um controlador PID e quatro arquiteturas para CLN foram implementadas.

Por fim, no capítulo 4, estão as conclusões do trabalho, onde estão descritas e discutidas as impressões do autor sobre as metas e os objetivos alcançados, como também a sugestão de tópicos para desenvolvimentos futuros.

Ao final desta dissertação encontra-se a bibliografia especializada utilizada para o desenvolvimento deste trabalho e em seguida poderão ser encontrados os códigos referentes aos algoritmos Nebulosos, os quais formam a base para a construção de controlador baseado em Lógica Nebulosa. As rotinas escritas em C, propostas por VIOT (1993), podem ser facilmente ampliadas e adaptadas para um maior número de Conjuntos Nebulosos ou Variáveis Lingüísticas, neste trabalho a função de nebulização teve o seu algoritmo otimizado.

CAPÍTULO 1

1 MODELAGEM DE SISTEMAS SERVOPNEUMÁTICOS

Este capítulo será dedicado à escolha do modelo a ser adotado para a representação do sistema servopneumático objeto deste trabalho de dissertação. A discussão sobre a opção por uma modelagem linear ou não-linear será tratada de forma clara salientando-se a opção seguida para desenvolvimento deste trabalho.

1.1 CARACTERIZAÇÃO DE UM SISTEMA SERVOPNEUMÁTICO

Partindo de um enfoque voltado à análise de um sistema dinâmico para a obtenção de leis de controle, pode-se caracterizar um sistema servopneumático como sendo não-linear e variante no tempo. Ou seja, as equações matemáticas que descrevem este sistema são equações diferenciais paramétricas, não-lineares, que incorporam fatores de bandas de passagem (atenuações e ruídos nos sinais elétricos), saturação, zonas mortas (ex. núcleos das servo válvulas supercríticos ou subcríticos) e atritos não-“newtonianos”, que são não-linearidades características de sistemas pneumáticos reais. Por sua vez, estes parâmetros internos do sistema podem variar no decorrer do tempo, devido aos desgastes causados por movimentos relativos de componentes mecânicos, colapso dos sistemas elétricos acoplados

e principalmente variações de carregamento, os quais provocam alterações significativas em parâmetros determinantes da dinâmica do sistema, como amortecimentos e frequências naturais. (SANTOS,1996); (VIEIRA ,1998)

Conforme exemplificado por ANDERSEN (1967), a própria formulação matemática para a modelagem de um escoamento compressível é não-linear, já que são empregadas as equações de estado e de escoamento em orifícios. Estas últimas, equações de natureza não-linear, devido, dentre outras coisas, às vazões variarem com a raiz quadrada do diferencial de pressão (numa análise simplista apenas para comparação). Esta modelagem apresenta-se dependente de coeficientes empíricos, que, a depender da análise, pode inviabilizar a sua utilização prática tanto pelo esforço, como pela difração da precisão do modelo devido aos parâmetros experimentais, tais como atrito e amortecimento.

Segundo CHEN et al. (1993), em um sistema servopneumático, a baixa rigidez, provocada pela compressibilidade do ar aliada às não-linearidades inerentes ao sistema dificultam as ações de controle.

É válido resaltar que, existe também a possibilidade de utilizar-se ferramentas de linearização destes sistemas. Apesar de tratar-se de uma análise pontual e, deste modo, não representativa para todo o universo em questão, a depender do objetivo que se pretenda alcançar, poderá vir a tornar-se uma modelagem funcional, prática e robusta.

Num modelo linearizado a atenção deve ser voltada para a seleção do ponto em torno do qual se realizarão as linearizações das equações que regem o sistema.

1.2 ANÁLISE NÃO-LINEAR DO SISTEMA SERVOPENUMÁTICO

Na definição do modelo matemático para servir como objeto de estudo desta dissertação, foi tomado bastante cuidado na escolha da abordagem a ser utilizada. Foram enumerados critérios que servissem de balizadores para esta tomada de decisão. Deveriam ser levados em conta aspectos de funcionalidade, flexibilidade, coerência e, principalmente, que o modelo deveria reproduzir com fidelidade e simplicidade adequada, o sistema real que lhe caiba representar, mesmo que pontualmente.

Propostas de modelos não-lineares podem ser encontradas no trabalho de dissertação de mestrado de SANTOS(1996) e VIEIRA(1998) realizados no Laboratório de Sistemas Hidráulicos e Pneumáticos do Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina (LASHIP). Dentre as modelagens presentes na literatura, consta uma aplicação executada por VIRVALO (1988), onde são trabalhadas as não-linearidades mais importantes do sistema: a força de atrito, a variação da frequência natural e os escoamentos compressíveis, estes últimos divididos em faixas sônica e subsônica.

Esta divisão em regiões de escoamento sônico e subsônico é onde reside a principal fundamentação de VIRVALO (1988) para o seu modelo não-linear.

1.2.1 Sistema de Equações

Considerando o sistema servo pneumático da Figura (1), onde estão representados o pistão, a massa útil e as câmaras do cilindro. Observa-se a delimitação da superfície e volume de controle (linha ---), englobando as câmaras do cilindro e as mangueiras de alimentação/descarga. A massa do sistema esta representada por "M", cujo valor engloba a massa transportada e a massa do pistão. As pressões nas câmaras são dadas em valores absolutos e representadas por P_1 e P_2 . A área útil de cada lado do pistão é dada por A_1 e A_2 . A massa e o volume morto de ar em cada câmara é dado por $m_{1/2}$ e $V_{10/20}$, respectivamente .

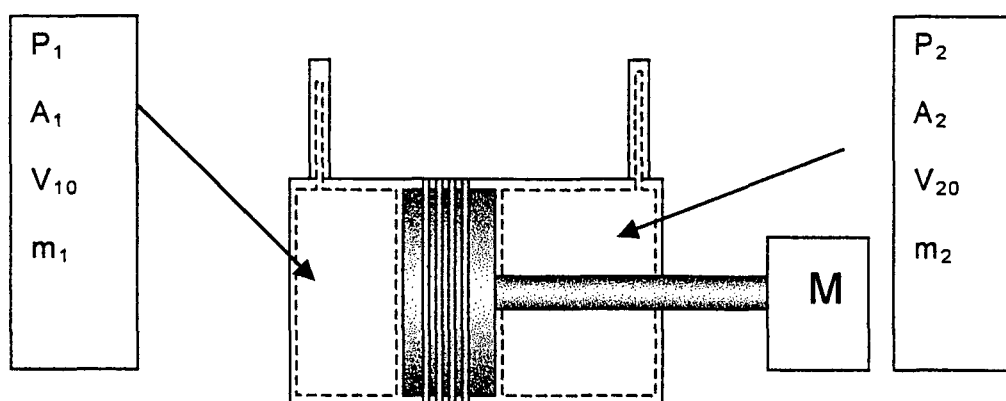


Figura 1- Variáveis de um sistema servo pneumático.

Tomando por base a equação da continuidade aplicada nas duas câmaras do sistema, incluindo os volumes mortos das mangueiras, tem-se:

$$\dot{p}_1 = \frac{\gamma}{V_{10} + A_1(1-x)}(\dot{m}_1 RT - p_1 A_1 \dot{x}) \quad (1)$$

$$\dot{p}_2 = \frac{\gamma}{V_{20} + A_2(1-x)}(\dot{m}_2 RT - p_2 A_2 \dot{x}) \quad (2)$$

Aplicando-se a segunda lei de Newton ao sistema do cilindro+massa com atrito obtém-se a Equação (3).

$$\ddot{x} = \frac{1}{M}[(p_1 - p_{at})A_1 - (p_2 - p_{at})A_2 - F_r - F_L] \quad (3)$$

Utilizando-se duas equações para modelar o escoamento compressível através de um orifício. Sendo que, a Equação (4) modela o escoamento e a Equação (5) é a função que corrige a vazão mássica para o escoamento supersônico, onde b representa a razão de pressão crítica.

$$\dot{m}_{1,2} = C p_s y \psi \quad (4)$$

$$\psi\left(\frac{p_1}{p_s}\right) = \begin{cases} \sqrt{1 - \left(\frac{\frac{p_1}{p_s} - b}{1 - b}\right)^2}, & \frac{p_1}{p_s} > b \\ 1, & \frac{p_1}{p_s} \leq b \end{cases} \quad (5)$$

A função Ψ pode ser representada graficamente como na Figura (2), onde podem ser observadas as zonas subsônicas e supersônicas.

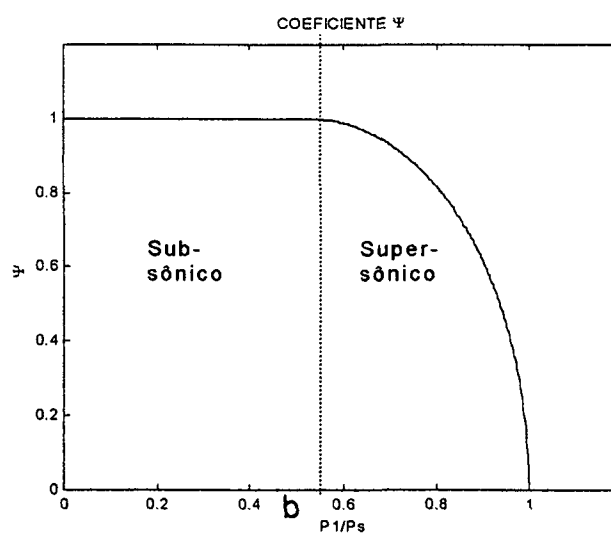


Figura 2- Coeficiente Ψ para correção da vazão mássica.

Para o deslocamento do carretel da servoválvula VIRVALO (1988) sugere utilizar uma aproximação de 2ª Ordem, representada na Equação (6).

$$\ddot{y} = k\omega^2 u + 2\zeta\omega\dot{y} + \omega^2 y \quad (6)$$

1.2.4 Determinação da razão de pressão crítica “b”

Considerando uma condição de estagnação em um escoamento isoentrópico, segundo VAN WILEN (1995), é possível escrever a relação de entalpia, entalpia de estagnação e a energia cinética como :

$$h + \frac{V^2}{2} = h_o \quad (7)$$

E em se tratando de escoamentos de gases perfeitos apresentando calor específico constante em todo volume de controle, a Equação (7) pode ser escrita como:

$$V^2 = 2c_{p_o}(T_o - T) = 2\frac{kRT}{k-1}\left(\frac{T_o}{T} - 1\right) \quad (8)$$

$$c^2 = kRT \quad (9)$$

Dividindo a Equação (8) pela Equação (9), obtém-se :

$$\frac{V^2}{c^2} = M^2 = \frac{2}{k-1}\left(\frac{T_o}{T} - 1\right) \quad (10)$$

Manipulando-se a Equação (10) pode-se dizer que:

$$\frac{T_o}{T} = 1 + \frac{k-1}{2} M^2 \quad (11)$$

Assumindo que para um processo isoentrópico, tem-se:

$$\frac{P_o}{P} = \left(\frac{T_o}{T} \right)^{\frac{k}{k-1}} \quad (12)$$

Desta forma, sabendo-se então que, quando a velocidade do escoamento é igual a velocidade do som no meio ($M=1$), trata-se da condição de vazão mássica máxima e substituindo a Equação (11) na Equação (12). Tem-se que a relação de pressão crítica pode ser definida por:

$$\frac{P}{P_o} = \left(\frac{2}{k+1} \right)^{\frac{k}{k-1}} \quad (13)$$

Considerado o ar, gás perfeito em escoamento isoentrópico, $k=1,4$, desta forma, tem-se então a razão de pressão crítica:

$$\frac{P}{P_o} = b = 0,5283 \quad (14)$$

1.2.3 Análise qualitativa do modelo não-linear de Virvalo (1988)

Este sistema de equações apresentado nas seções anteriores forma a modelagem não-linear proposta por VIRVALO (1988), alguns coeficientes das equações presentes neste modelo podem ser obtidos com o fabricante do componente (coeficiente de descarga da servoválvula por exemplo). No entanto, outros parâmetros devem ser ensaiados ou estimados (amortecimento do cilindro e da servoválvula). Admite-se utilizar servoválvulas com a dinâmica muito mais alta que o cilindro ($\omega_n > 150$ Hz), de forma que os sistemas permaneçam desacoplados.

Segundo VIRVALO (1988) esta formulação matemática satisfaz os requisitos de uma modelagem com fins de controle e sintonia de controladores para sistemas servopneumáticos. Contudo, realizando uma avaliação sob a ótica da função a qual estaria destinada a aplicação do modelo e pensando nos esforços necessários e benefícios atingidos, chega-se à conclusão que esta abordagem requer de esforços computacionais além do necessário já que uma modelagem mais simples, de acordo com o próprio VIRVALO (1989), pode representar o sistema satisfatoriamente.

1.3 ANÁLISE LINEARIZADA DO SISTEMA SERVOPNEUMÁTICO

Uma modelagem mais simples traria ganhos em termos de processamento matemático, facilitando assim a realização de simulações e análises em um maior número de aplicações de controle mas, em contrapartida, estariam limitadas as representações de não-linearidades (zonas mortas e atritos não-newtonianos).

VIRVALO (1989)¹, em outro artigo, realiza um paralelo entre a servohidráulica e a servopneumática, a fim de construir um modelo de terceira ordem simples e funcional. Procurando posicionar o ponto de linearização na condição de comportamento dinâmico mais adverso. Em seu artigo, VIRVALO (1989) atesta os bons resultados obtidos utilizando a sua modelagem linear de terceira ordem e recomenda esta abordagem para o projeto de controladores para posicionadores servopneumáticos. Resultados semelhantes foram obtidos por VIEIRA (1998) nos seus experimentos, no entanto, realizando ressalvas devido a necessidade de ajuste de parâmetros no modelo, no caso em questão, o amortecimento.

¹ Em dois artigos cronologicamente distintos Virvaio, um em 1988 e o outro em 1989, realiza para sistemas pneumáticos, uma modelagem não-linear e uma linear, no primeiro utiliza uma modelagem não-linear e constata a sua eficiência para a análise de sistemas pneumáticos, no seu segundo ensaio realiza uma modelagem linear simplificada e chega à mesma conclusão.

Para a formulação do modelo linear de terceira ordem é preciso que se anteceda a discussão sobre as condições de contorno sobre as quais o modelo deverá estar. De acordo com ZUMBRÄREL (1990), em primeiro plano, deve-se enumerar as hipóteses que fundamentam as simplificações e linearizações.

As hipóteses básicas são as seguintes:

- *A servo válvula deve possuir comportamento dinâmico linear, com frequências naturais muito acima do cilindro, podendo ser aproximada para um ganho de vazão.*
- *A servo válvula deve possuir centro crítico, não possuir histerese e zona morta.*
- *As mudanças de estado que ocorrem no sistema são isotérmicas e isoentrópicas.*
- *módulo de elasticidade volumétrica do ar (β) é constante e invariante no tempo.*
- *A pressão de alimentação (P_s) é constante.*
- *As câmaras do cilindro são simétricas ($\partial P_1 = -\partial P_2$).*
- *As áreas úteis no cilindro são iguais.*
- *As forças de atrito são lineares e estarão embutidas no coeficiente de amortecimento.*

A aproximação do modelo à realidade passa pelo atendimento destas premissas, nisto a seleção dos componentes do sistema tem papel fundamental. A utilização de válvulas críticas demandam um gasto maior na aquisição do componente, devido a precisão necessária para fabricação deste

componente. O atendimento à geometria do cilindro não é problema, mas esbarra na função que este cilindro vai atuar. O escoamento do ar no sistema ocorre em pressões baixas, a ponto que a consideração do ar como gás perfeito e c_p/c_v constante é uma aproximação razoável. A manutenção da pressão de suprimento constante demanda um gasto maior com a instalação componentes e controladores mais eficientes. A presença do atrito não-newtoniano nas instalações reais, representa um importante desvio sobre as premissas citadas. No entanto, em instalações onde as partes móveis estejam bem acabadas e lubrificadas, a distorção causada pelo atrito não-linear é minimizada.

Na Figura (3) encontra-se a forma esquemática e usual de um sistema servoposicionador pneumático para experimentos em bancada. Onde estão representados seus componentes básicos, como, cilindro, massa (carga+cilindro), servoválvula direcional e sistema de aquisição de dados e controle.

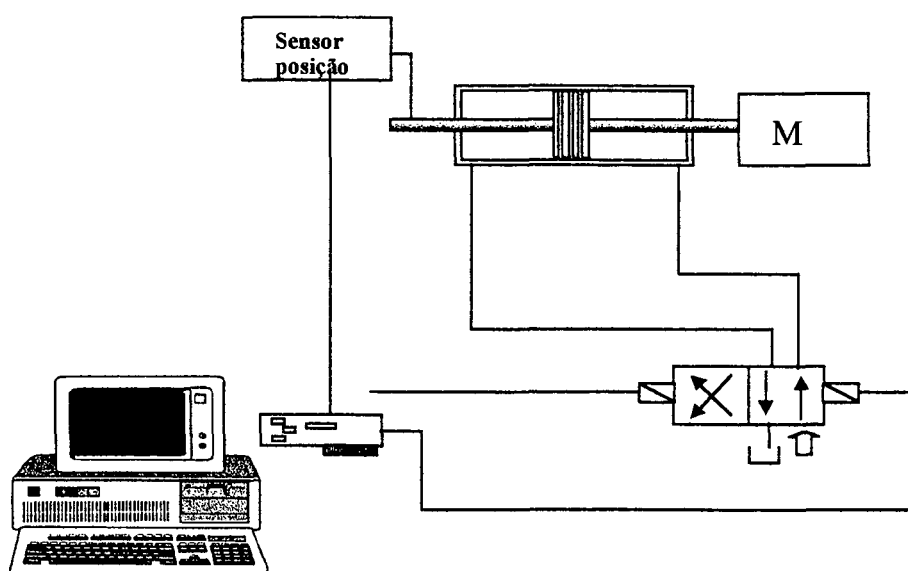


Figura 3 – Exemplo de um diagrama de um sistema pneumático usual.

1.3.1 Módulo de Elasticidade Volumétrica

Um dos principais parâmetros para a avaliação linearizada de um sistema servopneumático é o módulo de elasticidade volumétrica, já que este é o parâmetro responsável pela representação da compressibilidade do ar. O módulo de elasticidade volumétrica é definido como sendo a taxa de variação do volume em função da variação da pressão.

Por definição pode-se escrevê-lo como sendo:

$$\beta = -V \frac{\partial V}{\partial P} \quad (15)$$

Quando se refere ao ar o " β_{ar} " passa a ser:

$$\beta_{ar} = \gamma P \quad (16)$$

É possível demonstrar o valor de " β_{ar} " de duas formas, uma partindo das propriedades de um gás perfeito e a outra a partir da definição de um processo isoentrópico.

Segundo VAN WYLEN (1995), para o ar considerado como gás perfeito e em condições isoentrópicas tem-se :

$$\gamma = \frac{c_{po}}{c_{vo}} \quad (17)$$

$$c_{vo} = \frac{R}{\gamma - 1} \quad (18)$$

$$c_{po} = \frac{\gamma R}{\gamma - 1} \quad (19)$$

Existe a seguinte relação termodinâmica entre as propriedades:

$$Tds = du + pdv \quad (20)$$

Como para um processo adiabático e reversível tem-se que $ds=0$ a Equação (20) torna-se:

$$du + pdv = 0 \quad (21)$$

$$du = c_{vo} dT \quad (22)$$

Substituindo a Equação (22) na Equação (21), obtem-se:

$$c_{vo} dT + pdv = 0 \quad (23)$$

Como para os Gases Perfeitos é válida a relação entre Pressão, Volume e Temperatura encontrada na Equação (24).

$$Pv = RT \quad (24)$$

Derivando-se a Equação (24):

$$dPv + Pdv = RdT \quad (25)$$

Isolando dT:

$$dT = \frac{1}{R} + (pdV + vdp) \quad (26)$$

Combinando a Equação (25) e (26).

$$\frac{c_{vo}}{R} (pdv + vdp) + Pdv = 0 \quad (27)$$

Substituindo a Equação (18) na Equação (27) chega-se a:

$$\frac{1}{\gamma - 1} (pdv + vdp) + Pdv = 0 \quad (28)$$

e rearranjando a Equação (28):

$$\beta_{ar} = v \frac{dp}{dv} = -\gamma p \quad (29)$$

onde:

γ = coeficiente isoentrópico . Para o ar 1,4.

Alternativamente pode-se demonstrar o valor de " β_{ar} " de uma outra forma utilizando a definição para um processo isoentrópico.

Como o processo é isoentrópico, pode-se escrever que (SMITH e VAN NESS, 1949):

$$pv^\gamma = cte \quad (30)$$

Derivando-se a Equação (31):

$$dpv^\gamma + pv^{\gamma-1}dv = 0 \quad (31)$$

$$\beta = -v \frac{dp}{dv} = \gamma p \quad (32)$$

1.3.2 Frequência Natural

Dando prosseguimento a construção do modelo linear, de acordo com a analogia entre a hidráulica e pneumática, equaciona-se a frequência natural no sistema servopneumático.

Segundo BOLLMANN (1994) e VIRVALO (1989) pode-se considerar que, após um sinal de entrada impulso (ex. golpe na haste), a oscilação da posição de referência de um cilindro pneumático com os seus pórticos de entrada bloqueados, se deve ao efeito mola presente nos fluidos compressíveis.

Desta forma é também possível traçar uma analogia do sistema servopneumático a um sistema massa+mola. A Figura (4) ilustra as variáveis análogas entre estes dois sistemas.

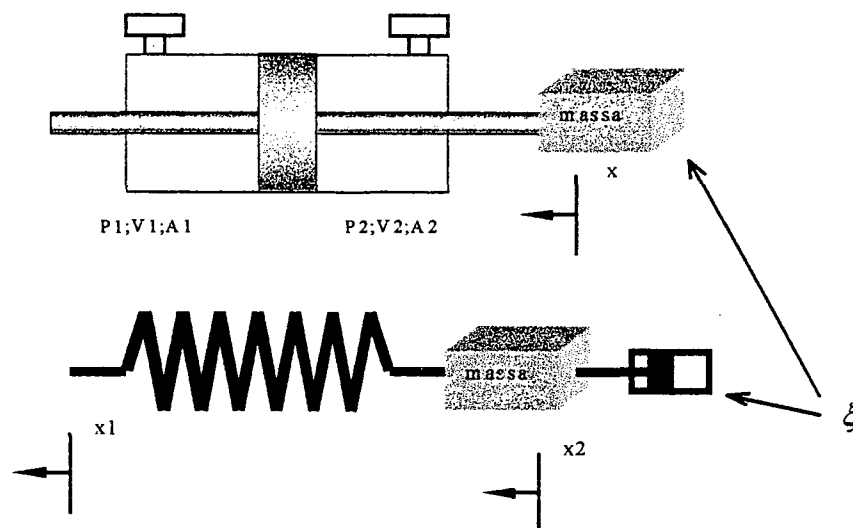


Figura 4 – Representação de um sistema massa+mola clássico e um sistema servopneumático.

Considerando primeiro o sistema pneumático, se faz necessário observar algumas condições de contorno para que o desenvolvimento do equacionamento possa ser continuado:

- São necessárias todas as premissas anteriores para a formulação de um modelo matemático linear;
- cilindro é de duplo efeito, sendo $A_1=A_2=A$;
- As câmaras V1 e V2 inicialmente estão na mesma pressão inicial;
- sistema parte da referência $x=0$, como ponto inicial para início do movimento;
- pistão parte do repouso (velocidade inicial=0);
- Pórticos de entrada e saída bloqueados;

Aplicando-se um impulso ao sistema ocorre inicialmente uma variação simétrica de pressão nas câmaras do cilindro e em decorrência disto, em conjunto com o efeito mola, o sistema retorna para a posição simetricamente oposta, mantendo um movimento oscilatório perpétuo. Nesta condição pode-se obter a frequência natural do sistema.

Avaliando as condições em cada uma das câmaras do cilindro:

Pressões:

$$P_1 = P + \Delta P_1 \quad (33)$$

$$P_1 = P + \frac{\beta}{V_1} A \ x \quad (34)$$

$$P_2 = P + \Delta P_2 \quad (35)$$

$$P_2 = P - \frac{\beta}{V_2} A \ x \quad (36)$$

Subtraindo-se a Equação (34) da Equação (35):

$$(P_1 - P_2) = \beta A \left(\frac{1}{V_1} + \frac{1}{V_2} \right) x \quad (37)$$

Aplicando a Equação do Movimento ao cilindro:

$$-(P_1 - P_2)A = M \frac{d^2 x}{dt^2} \quad (38)$$

Substituindo a Equação (37) na Equação (38),

$$\frac{1}{\beta \left(\frac{1}{V_1} + \frac{1}{V_2} \right)} \frac{M}{A^2} \frac{d^2 x}{dt^2} + x = 0 \quad (39)$$

Para o sistema massa mola tem-se:

$$M \frac{d^2 x_2}{dt^2} = -kx_2 \quad (40)$$

$$\frac{M}{k} \frac{d^2 x_2}{dt^2} + x_2 = 0 \quad (41)$$

Um sistema oscilatório sem amortecimento e sem força excitatriz em a forma:

$$\frac{1}{\omega_n^2} \frac{d^2 x_2}{dt^2} + x_2 = 0 \quad (42)$$

Onde:

Para massa-mola:

$$\omega_n = \sqrt{\frac{k}{M}} \quad (43)$$

Para pneumático:

$$\omega_n = \sqrt{\frac{\beta \left(\frac{1}{V_1} + \frac{1}{V_2} \right) A^2}{M}} \quad (44)$$

Nesta modelagem observar-se que um dos elemento passíveis de variação é a frequência natural do sistema. Esta varia com a posição do cilindro no decorrer do seu curso e com a carga. A linearização deste sistema é dependente da atribuição de um valor constante para esta frequência. É preciso definir então, qual será o valor mais representativo para esta, tendo em vista a sua importância para a efetividade do controle sintonizado e para a estabilidade do sistema.

A escolha do ponto em torno do qual o sistema será linearizado, é dependente do comportamento dinâmico do sistema neste ponto. No sistema objeto deste trabalho optou-se pelo ponto mais desfavorável para o sistema, o ponto onde a frequência natural é mínima. Desta forma garante-se que o controle sintonizado para esta condição vai ser satisfatório para qualquer outra condição. (STRINGER, 1976)

Analogamente a um sistema hidráulico, conforme afirmado por STRINGER (1976), para que a frequência natural do sistema pneumático seja mínima é preciso que os volumes V_1 e V_2 sejam iguais.

Desta forma:

$$V_2 = V_1 = \frac{V_t}{2} \quad (45)$$

E a Equação (44) pode ser rescrita como:

$$\omega_n = \sqrt{\frac{4\beta A^2}{V_t M}} \quad (46)$$

Para demonstrar esta afirmação utiliza-se o método dos máximos e mínimos para funções de múltiplas variáveis condicionadas de Lagrange, de acordo com BRONSTEIN et al. (1979), como segue:

Para que:

$$\omega_n \approx \omega_{n\text{mínimo}} \quad (47)$$

É preciso que:

$$\left(\frac{1}{V_1} + \frac{1}{V_2} \right)_{\text{mínimo}} \quad (48)$$

Usando o sistema de Lagrange:

$$f(V_1, V_2) = \left(\frac{1}{V_1} + \frac{1}{V_2} \right) \quad (49)$$

$$\Theta(V_1, V_2) = V_1 + V_2 - V_t = 0 \quad (50)$$

$$\frac{\partial}{\partial V_1} \left[\frac{1}{V_1} + \frac{1}{V_2} + \lambda(V_1 + V_2 - V_t) \right] = 0 \quad (51)$$

$$\frac{\partial}{\partial V_2} \left[\frac{1}{V_1} + \frac{1}{V_2} + \lambda(V_1 + V_2 - V_t) \right] = 0 \quad (52)$$

Resolvendo o sistema:

$$V_2 = V_1 = \frac{V_t}{2} \quad (53)$$

Observando a vizinhança deste ponto médio, constata-se que este, realmente, trata-se de um ponto de mínimo.

1.3.3 Sistema de Equações Completo

Equacionamento extraído de BOLLMANN(1994) e STRINGER (1996).

Na servo-válvula:

Vazão de ar:

$$Q_C = f(X_1, P_L) \quad (54)$$

Expandindo a equação acima em série de Taylor, tem-se entorno do ponto '1', que seria o ponto central da válvula:

$$Q_C = Q_{C1} + \left. \frac{\partial Q_C}{\partial X_1} \right|_1 \Delta X_1 + \left. \frac{\partial Q_C}{\partial P_L} \right|_1 \Delta P_L + \dots \quad (55)$$

Desprezando-se os termos de ordem superior da série:

$$Q_C = Q_{C1} + \left. \frac{\partial Q_c}{\partial X_1} \right|_1 \Delta X_1 + \left. \frac{\partial Q_c}{\partial P_L} \right|_1 \Delta P_L \quad (56)$$

$$(Q_C - Q_{C1}) = \left. \frac{\partial Q_c}{\partial X_1} \right|_1 \Delta X_1 + \left. \frac{\partial Q_c}{\partial P_L} \right|_1 \Delta P_L \quad (57)$$

$$\Delta Q_C = \left. \frac{\partial Q_c}{\partial X_1} \right|_1 \Delta X_1 + \left. \frac{\partial Q_c}{\partial P_L} \right|_1 \Delta P_L \quad (58)$$

Coeficiente de vazão:

$$K_q = \left. \frac{\partial Q_c}{\partial X_1} \right|_1 \quad (59)$$

Coeficiente de pressão:

$$K_C = \left. \frac{\partial Q_c}{\partial P_L} \right|_1 \quad (60)$$

Para variações em torno de um único ponto referencial:

$$Q_C = K_q X_1 + K_C P_L \quad (61)$$

Para válvulas sem vazamentos.

$$Q_C = K_q X_1 \quad (62)$$

No cilindro:

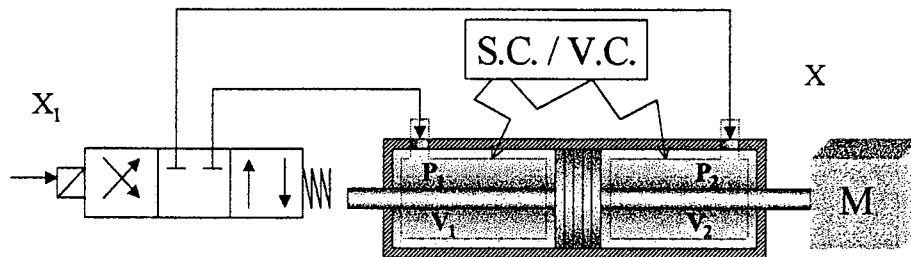


Figura 5 – Identificação do volume de controle e da superfície de controle.

Equação Geral da continuidade:

$$\frac{\partial}{\partial t} \oint_C \rho \partial v = \oint_{SC} \rho \vec{S} \vec{n} \partial A \quad (63)$$

Na Superfície de Controle :

$$\oint_{SC} \rho \vec{S} \vec{n} \partial A = \quad (64)$$

$$\rho = \text{Constante}$$

$$\vec{S} = \text{Constante}$$

Sendo que os valores de ρ e \vec{S} podem variar no tempo mas sobre a superfície de controle podem ser considerados constantes, já que o escoamento é laminar adiabático e sem atritos.

$$\rho \oint_{SC} \vec{S} \vec{n} \partial A = \quad (65)$$

$$\rho(S_e A_e - S_s A_s) = \quad (66)$$

$$\rho(Q_e - Q_s) = \quad (67)$$

No Volume de Controle :

$$= \frac{\partial}{\partial t} \oint_{\text{VC}} \rho \partial v \quad (68)$$

$$= \frac{\partial}{\partial t} (\rho V) \quad (69)$$

$$= \rho \frac{\partial V}{\partial t} + V \frac{\partial \rho}{\partial t} \quad (70)$$

$$= \rho \frac{\partial V}{\partial t} + V \frac{\partial \rho}{\partial P} \frac{\partial P}{\partial t} \quad (71)$$

$$\beta = -V \frac{\partial P}{\partial V} \quad (72)$$

$$V = M / \rho \quad (73)$$

$$\partial V = -M / \rho^2 \partial \rho \quad (74)$$

$$= \rho \frac{\partial V}{\partial t} + V \frac{\rho}{\beta} \frac{\partial P}{\partial t} \quad (75)$$

$$(Q_e - Q_s) = \frac{\partial V}{\partial t} + \frac{V}{\beta} \frac{\partial P}{\partial t} \quad (76)$$

Equação do movimento:

$$(P_1 - P_2)A - f \frac{\partial X}{\partial t} = M \frac{\partial^2 X}{\partial t^2} \quad (77)$$

Sabendo que:

$$\frac{\partial V}{\partial t} = A \frac{\partial X}{\partial t} \quad (78)$$

Pela servoválvula ter centro crítico e simétrico:

$$\partial P_1 = -\partial P_2 \quad (79)$$

$$P_L = P_1 - P_2 \quad (80)$$

$$-\partial P_2 = \partial P_1 = \frac{\partial P_L}{2} \quad (81)$$

$$Q_c = Q_1 = Q_2 \quad (82)$$

Ponto mais desfavorável:

$$V_1 = V_2 = \frac{V_T}{2} \quad (83)$$

Aplicando a eq. da continuidade câmara 1:

$$(Q_1) = A \frac{\partial X}{\partial t} + \frac{V_1}{\beta} \frac{\partial P_1}{\partial t} \quad (84)$$

$$Q_c = A \frac{\partial X}{\partial t} + \frac{V_T}{4\beta} \frac{\partial P_L}{\partial t} \quad (85)$$

Fazendo-se as devidas substituições:

Em transformada de Laplace:

$$\frac{X(s)}{X_{e(s)}} = \frac{K_q / A}{s \left(\frac{V_t}{4\beta} \frac{M}{A^2} s^2 + \frac{V_t f}{4\beta A^2} s + 1 \right)} \quad (86)$$

A Figura (6) ilustra a função transferência deste sistema:

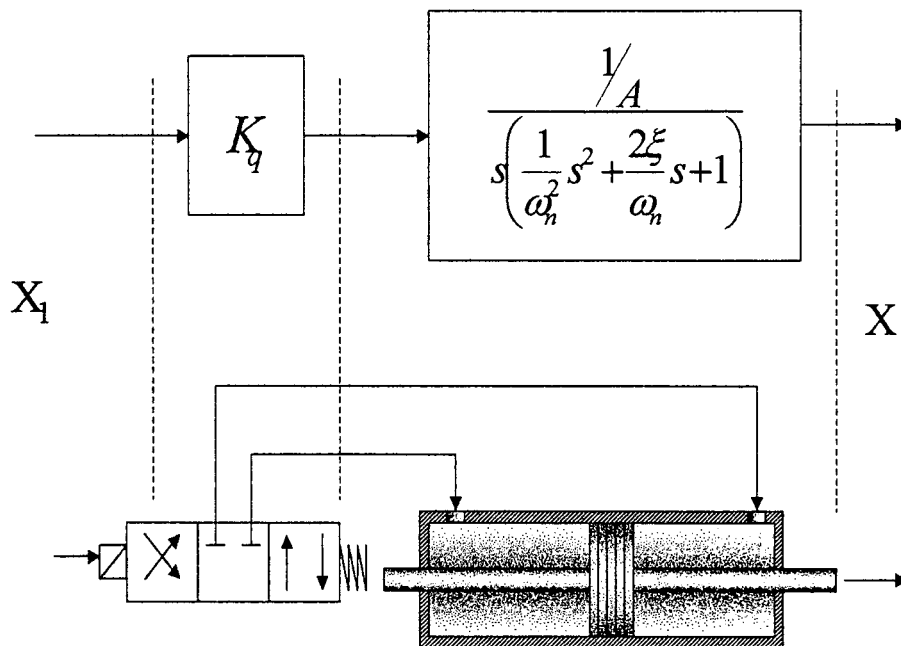


Figura 6 – Sistema servopneumático e sua função de transferência linear de terceira ordem.

1.3.4 Servoválvula e Amortecimento

Em um sistema servopneumático, segundo VIRVALO (1989), a banda de frequência natural de uma servoválvula eletropneumática (100 à 150 Hz) é

significativamente mais alta que a do cilindro (1 à 10 Hz), desta forma pode-se considerar que os sistemas estão desacoplados e, por isso, reduzir a dinâmica da servoválvula para um ganho (K_q), um sistema de zero grau de ordem.

Para a análise e avaliação do K_q , o coeficiente de descarga da servo válvula, deve-se estabelecer mais algumas restrições ao contorno do sistema. Dentre elas, as mais importantes são a ausência de vazamentos e a constância da pressão de suprimento de ar para o sistema.

Além disto se faz necessário caracterizar o sistema, assumindo que:

- *A queda de pressão sobre a servoválvula é igual a 1/3 P_s .*
- *A pressão no interior do cilindro é 2/3 P_s*
- *Q_N é a vazão Normal da servoválvula para uma queda de pressão de 1 bar. Q_N é um dado fornecido pelo fabricante da servoválvula.*

Como a vazão de um fluido compressível é função de uma constante e a raiz quadrada da diferença de pressão, então:

$$Q \approx \phi \sqrt{\Delta P} \quad (87)$$

Por definição Q_N (ISO, 1989) vai ser função de :

$$Q_N \approx \phi \sqrt{1} \quad (88)$$

Mas para o sistema em questão a queda de pressão é $1/3 P_s$, daí tem-se que a vazão Q_N' , será:

$$Q_N' \approx \varphi \sqrt{1/3 P_s} \quad (89)$$

$$\frac{Q_N'}{Q_N} \approx \frac{\varphi \sqrt{1/3 P_s}}{\varphi \sqrt{1}} \quad (90)$$

$$Q_N' \approx Q_N \sqrt{1/3 P_s} \quad (91)$$

Tem-se então o que seria a Vazão Volumétrica Normal para o sistema pneumático em questão ($1/3 P_s$ de queda de pressão sobre a servo-válvula).

No entanto, o que interessa para o modelo como variável de entrada, é a vazão de ar que entra no cilindro. Desta forma, utilizando-se da definição de vazão normal, pode-se calcular a vazão que entra no cilindro (pressão interna do cilindro igual $2/3 P_s$ manométrica):

$$Q_c = \frac{Q_N'}{\left(\frac{2}{3} P_s + 1 \right)} \quad (92)$$

$$Q_c = \frac{Q_N \sqrt{1/3 P_s}}{\left(\frac{2}{3} P_s + 1 \right)} \quad (93)$$

A partir da equação da vazão volumétrica de ar no cilindro pode-se então fazer a linearização do sistema da servoválvula para um ganho.

$$Q_C = K_q X_1 \quad (94)$$

$$K_q = \frac{Q_N \sqrt{1/3 P_s}}{\left(\frac{2}{3} P_s + 1\right) X_1} \quad (95)$$

A análise do amortecimento (ξ) num sistema servopneumático linearizado deve sofrer algumas considerações :

- *Este valor de amortecimento deve levar em consideração o atrito físico, o atrito viscoso e o amortecimento por perdas fluidas, se for considerado um sistema com vazamentos.*
- *Trata-se de um valor que sem a realização de medições empíricas não é possível precisar matematicamente. Estudos para aproximações para representação das forças de atrito foram realizadas por SANTOS (1996) e VIEIRA (1998).*
- *Na prática estes valores estão na faixa de 0.05 à 0.2. (VIRVALO, 1989)*
- *Baseado em artigo de VIRVALO (1989), a utilização de um valor de 0.15 para o amortecimento tem provado constituir-se numa boa aproximação. No entanto esta afirmativa não é aceita por VIEIRA (1998) no seu trabalho de dissertação.*

1.3.5 Análise Qualitativa do Modelo linearizado

De acordo com VIRVALO (1989) e VIEIRA (1998) o modelo linearizado de um sistema servopneumático mostrou-se adequado para a sua representação. E objetivando a sua utilização para simulações de perturbação de carga, tem-se os dois principais parâmetros, frequência natural e amortecimento, representados, dentre outros parâmetros, em função da carga. De alguma forma a compressibilidade do ar também é representada no modelo através do β . A saturação de sinal na servoválvula é uma não-linearidade importante para o modelo, esta estará representada por um bloco limitador externo ao modelo.

Vale ressaltar que este modelo pode ser refinado utilizando dados experimentais específicos, para que possa melhor representar o sistema que se deseja modelar. Utilizando-se este modelo como um núcleo, não-linearidades do tipo zona-morta, ruídos e saturações podem ser implementadas. Estas características aliada a facilidade de implementação do modelo, acabou por motivar a opção por uma abordagem linearizada para este trabalho.

CAPITULO 2

2 A LÓGICA NEBULOSA

Os avanços na ciência cognitiva resultaram no desenvolvimento de diferentes ferramentas para o seu suporte, algumas direcionadas para técnicas numéricas (ex.: redes neurais) e outras direcionadas para sistemas de inferência (ex.: lógica nebulosa e sistemas especialistas). A partir da formulação para tradução da razão humana através de variáveis lingüísticas obedientes a uma semântica e sintaxe particular, originou-se a Lógica Nebulosa, desenvolvida por Lofti A. Zadeh em 1965.

A fundamentação da Lógica Nebulosa foi lançada por ZADEH no seu artigo "FUZZY SET THEORY" (ZADEH,1965), onde foi desenvolvido o conceito de CONJUNTOS DIFUSOS ou CONJUNTOS NEBULOSOS (FUZZY SETS), definida a utilização das variáveis lingüísticas e sua sintaxe. Baseada nos conceitos de vaga e imprecisão, a Lógica Nebulosa procura codificar o raciocínio humano para aplicações no âmbito da tomada de decisão, do reconhecimento e da abstração. Tornando-se uma ferramenta importante para a resolução de problemas que não estão completamente claros e definidos. (LEE, 1990)

Ao longo do seu desenvolvimento foram surgindo aplicações diversas para esta ferramenta. Dentre elas, baseado na sua capacidade de tomada de decisão, a

Lógica Nebulosa passou a ser utilizada na construção de controladores inteligentes. (MANDANI,1974)

Dentre as diversas aplicações dos Controladores Baseados em Lógica Nebulosa, destacam-se na literatura:

- Caldeira. (MANDANI, 1974)
- Turbina à Vapor. (MANDANI e ASSILIAN, 1975)
- Forno de Cimento. (ZADEH,1984)
- ATO – Automação da Operação de Trens (ZADEH,1984)
- Controle de Direção de um Veículo (ZADEH,1984)
- Controle de Voo (LARKIN,1985)
- Motor Servoposicionador. (Li e LAU, 1989)
- Reator Nuclear (LEE,1990)
- Servoposicionador Hidráulico. (NIMELLA e VIRVALO ,1993)
- Turbina a Gás. (BONISSONE et al.,1995)
- Partida de Turbina a Vapor. (BONISSONE et al.,1995)
- Gerenciamento de Carga em Turbinas a Vapor em Termoelétricas. (BONISSONE et al.,1995)
- Regulador de Voltagem para Fornecimento de Energia Elétrica. (BONISSONE et al.,1995)
- Incinerador de Lixo. (ALTROCK,1999)
- Tratamento de água efluente. (ALTROCK,1999)

Industrialmente uma das primeiras aplicações de um controlador baseado em Lógica Nebulosa foi em um forno para cimento no ano de 1980 (ZADEH,1984). Esta foi a aplicação do primeiro microcontrolador dedicado baseado em Lógica Nebulosa, fabricado e desenvolvido pela F.L. Smidth Co. sediada em Compenhagen (Dinamarca).

Uma década após a primeira venda e aplicação de um controlador inteiramente Nebuloso, a empresa japonesa Omron Corp. desenvolveu um microprocessador Nebuloso e implementou como módulo de co-processamento para a sua linha de controladores lógicos programáveis (CLPs). Então, no ano de 1992, a Klockner-Moeller (Alemanha) projetou e pôs no mercado o primeiro CLP-Nebuloso dedicado. Só em 1994 a Allen-Bradley e a Siemens desenvolveram a suas primeiras soluções em Lógica Nebulosa, no entanto, apenas implementadas em software. Em 1997 a IEC *International Electrotechnical Commission* (IEC) criou a Norma *CD IEC 1131-7 fuzzy-logic* integrando a Lógica Nebulosa aos padrões de CPL existentes. (ALTROCK,1999)

Segundo ALTROCK(1999), em geral os controladores PID e o controle ON-OFF industriais são utilizados em malhas controle individuais, cabendo ao operador ajustar as referências manualmente a fim de otimizar o processo em cada ponto operacional. A Lógica Nebulosa possibilita ao Engenheiro de Controle desenvolver controladores supervisórios multivariáveis a partir do conhecimento dos operadores ou de resultados experimentais ao invés de basear-se em modelos matemáticos. Estes sistemas supervisórios são

aplicados às malhas de controle PID ou ON-OFF para que suas referências sejam ajustadas sem interferência do operador, buscando a otimização do processo nos diversos pontos operacionais.

De acordo com JANTZEN (1998d), controladores nebulosos de alto nível além de ajustar as referências, modificam os parâmetros internos dos controladores convencionais da planta, visando garantir estabilidade e otimização. Como também, em outra configuração, implementa-se um controlador nebuloso para atuar em situações de mudanças bruscas no processo. Neste caso, quando o sistema estiver em estado normal de operação a contribuição do controlador nebuloso é zero.

A Lógica Nebulosa cada vez mais se difunde no ambiente industrial e acadêmico, em áreas das mais diversas, como: Reconhecimento de imagens para classificação de células cancerígenas (BYRIEL, 1999); Controle de rotas para robôs autômatos (TAKEUCHI e FURUHASHI, 1999 e SANO e TAKAHASHI, 1994); modelagem de sinais (KOSANOVIC et al., 1996), controle supervisorio em processos variantes no tempo e multivariáveis (YAZDI, 1997) e outras mais.

2.1 CONTROLADOR BASEADO EM LÓGICA NEBULOSA (CLN)

Os elementos construtivos de um controlador baseado em Lógica Nebulosa estão bem definidos dentro do seu diagrama funcional. A construção do Controlador Nebuloso se distingue em três níveis funcionais básicos, são eles: duas interfaces com os valores não Nebulosos e uma base de conhecimentos.

De acordo LEE (1990), através dos conceitos de Lógica Nebulosa pode-se definir os elementos constitutivos de um controlador Nebuloso como sendo: NEBULIZAÇÃO, AVALIAÇÃO DE REGRAS e DESNEBULIZAÇÃO. Cada um destes elementos possui características funcionais e construtivas próprias. A Figura (7) representa esta arquitetura.

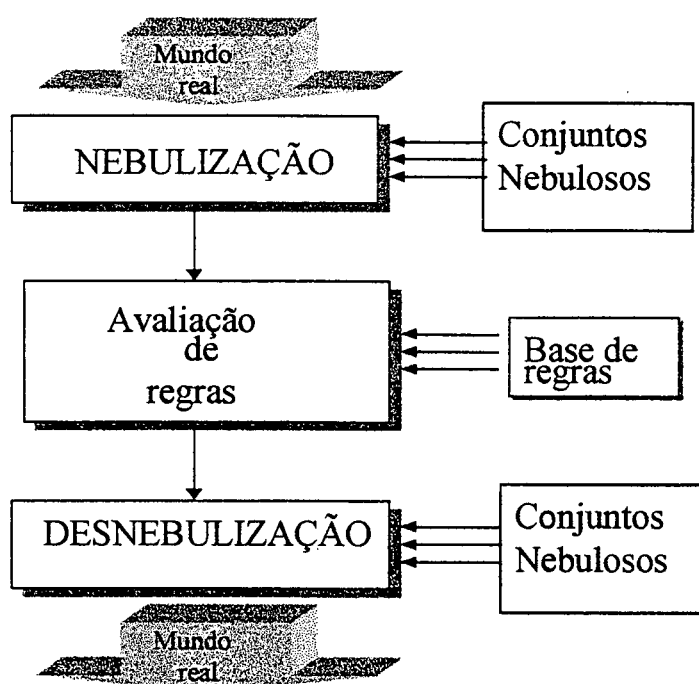


Figura 7 - Representação do processo de um controlador baseado em Lógica Nebulosa.

2.1.2 NEBULIZAÇÃO

A Nebulização transporta os valores absolutos medidos no meio físico, para uma linguagem abstrata, própria ao Raciocínio Nebuloso. Segundo LEE (1990) são funções da Nebulização:

- i. Medir os valores das variáveis de entrada.
- ii. Efetuar um mapeamento em escala (normalização) dos valores das variáveis de entrada para adequá-los ao respectivo universo de discurso.
- iii. Executar a função de Nebulização que converte os dados de entrada em valores lingüísticos pertencentes a variável lingüística correspondente.

Variáveis Lingüísticas. Segundo ZADEH, citado por MENDEL (1995), as Variáveis Lingüísticas podem ser conceituadas como sendo variáveis que ao invés do seus valores serem números, são palavras ou sentenças. Estas variáveis podem ser caracterizadas por um quinteto $(x, T(x), U, G, M)$, formado por:

- i. O nome da variável lingüística. (x)
- ii. Conjunto de Valores Lingüísticos. $(T(x))$
- iii. Universo de Discurso. (U)
- iv. A regra sintática para geração dos nomes dos valores lingüísticos. (G)
- v. A regra semântica que associa cada valor ao seu significado. (M)

Por exemplo:

Variável Lingüística (x): Velocidade

Conjunto de Valores Lingüísticos T(x): {devagar; moderado; rápido; muito rápido; mais ou menos devagar; ...}

Universo de Discurso (U): [0,100] km/h

Regra Sintática (G): “Os nomes dos valores devem ser apresentados na forma de simples (“ADJETIVOS”) ou modificada (“ADVERBIO+ADJETIVO”) . “

Regras Semânticas (M) : “ Valor (devagar) = [velocidades abaixo e entorno de 40 km/h]”

Conjunto Nebuloso. Um conjunto Nebuloso (F) é caracterizado por uma função de pertinência (μ_F) que assume valores no intervalo de [0,1], sobre um Universo de Discurso (U). Segundo LEE (1990) o Conjunto Nebuloso também pode ser visto como uma generalização do conceito da Lógica Clássica onde a função pertinência apenas assume valores {0,1}. Admitindo-se um elemento genérico (u) pertencente ao Universo de Discurso (U) a função de pertinência (μ_F), mapea o Conjunto Nebuloso (F) sobre o Universo de Discurso (U) para cada elemento (u). Tal que, o Conjunto Nebuloso (F) é representado no Universo de Discurso discreto (U_d) como um conjunto de pares ordenados.

$$F=\{(u, \mu_F(u))|u \in U_d\}.$$

Pontos significativos do Conjunto Nebuloso (LEE,1990):

Cruzamento – Pontos onde o grau de pertinência é 0,5.

Suporte – Conjunto de pontos onde o grau de pertinência é maior que 0.

Um conjunto Nebuloso e seus pontos significativos estão representados na Figura (8).

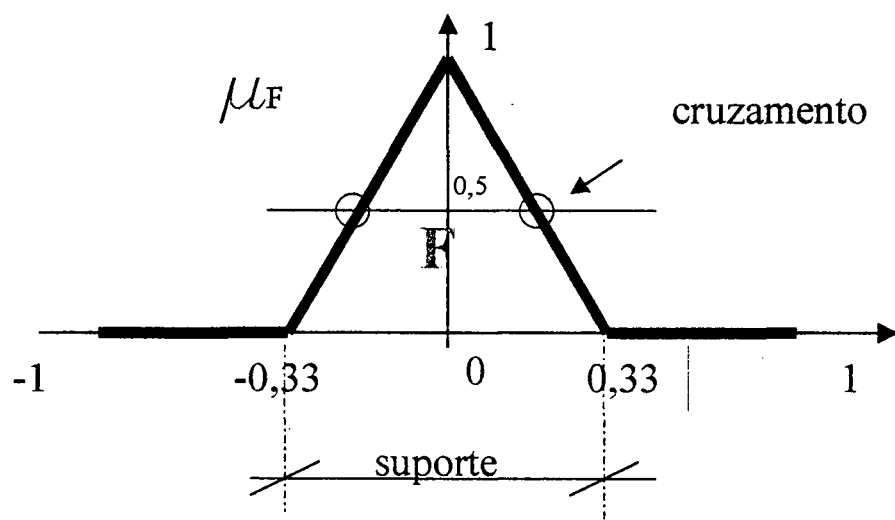


Figura 8 –Representação de um Conjunto Nebuloso e seus pontos significativos.

Universo de Discurso. O Universo de discurso é um conjunto de objetos discretos ou contínuos que compõem o universo de trabalho. Uma das tarefas da Nebulização é ajustar a faixa de medições das variáveis de entrada ou saída, para adequá-las ao Universo de Discurso. Geralmente os valores de entrada são normalizados para assumirem valores entre +1 e -1. Sobre o Universo de Discussão são distribuídos os Conjuntos Nebulosos, os quais possuem funções próprias que relacionam a variável lida ao seu Grau de Pertinência a um determinado Conjunto Nebuloso. O valores associados ao grau de pertinência variam entre 0 e 1.

Função de pertinência (μ_F). A Função Pertinência estabelece o grau de similaridade de um elemento do universo de discurso a um determinado Conjunto Nebuloso. A Função de Pertinência pode assumir diversas formas, dentre elas as mais comuns são a ríspida, unitária (*singleton*), monotônica crescente ou decrescente, trapezoidal, triangular e gaussiana. (COX, 1993, MENDEL, 1995 e NOBRE e PALHARES, 1997)

Na Figura (9) estão respectivamente representadas estas formas.

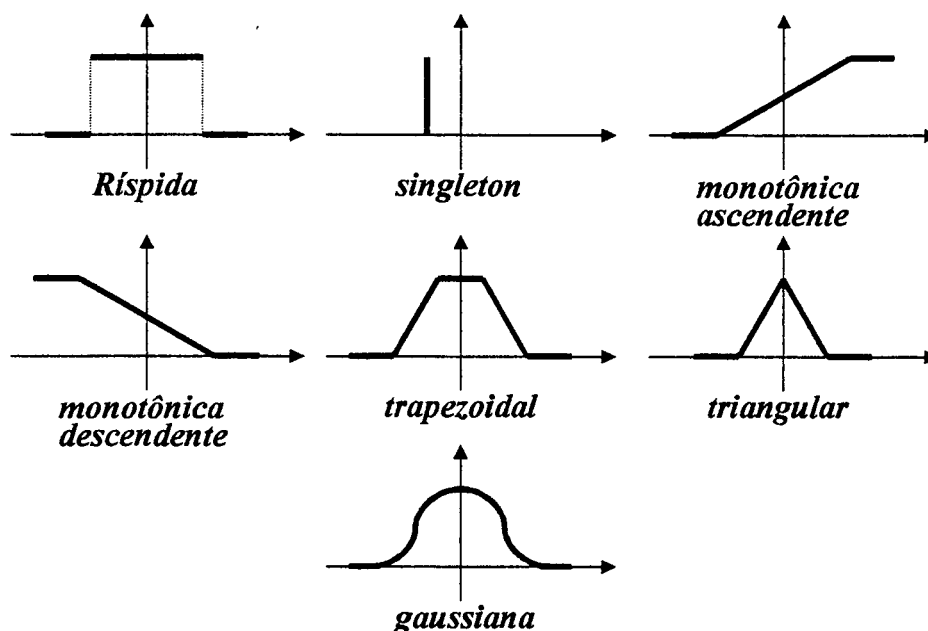


Figura 9 – Formas comuns para funções de pertinência.

O formato da Função de Pertinência em conjunto com Universo de Discurso formam a semântica das Variáveis Nebulosas. A flexibilidade na escolha, de qual função aplicar, permite que Variável Lingüística assumam diferentes

significados, para que esta possa refletir, na melhor forma, o universo ao qual ela corresponde. (SUGENO,1985)

Para fins de construção de um controlador baseado em Lógica Nebulosa recomenda-se que se inicie pela utilização das formas triangulares, devido a sua linearidade, facilidade de realização através de uma linguagem de programação e velocidade no momento da sua execução. (COX,1994 e JANTZEN,1998a)

A Figura (10) representa um mapa característico de uma Variável Linguística .

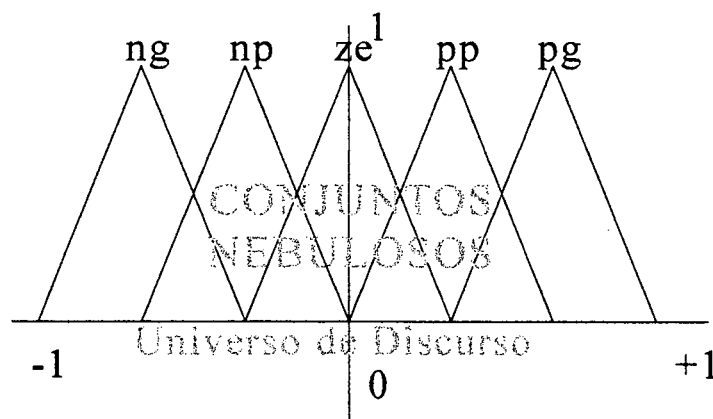


Figura 10 – Mapa de uma Variável Linguística em um processo de Nebulização.

Um exemplo comum de Nebulização de um valor de entrada é apresentado na Figura (11), onde, para a referida entrada, são encontrados diferentes Graus de Pertinência associados a diferentes Valores Lingüísticos.

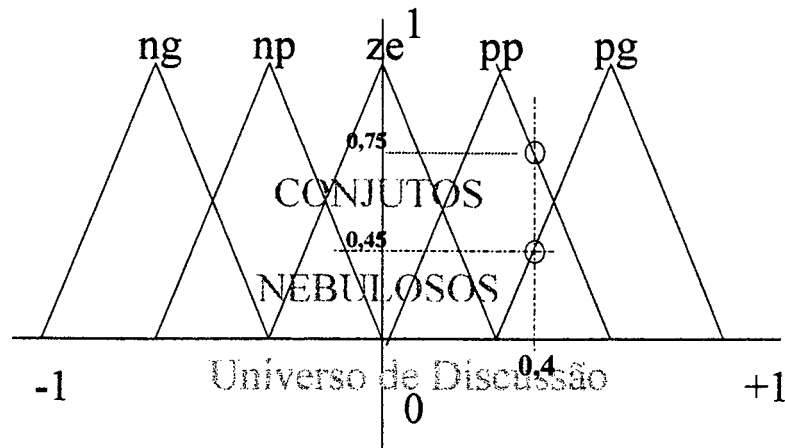


Figura 11 – Exemplo de entrada em um processo de Nebulização.

Obtendo-se como saída:

$$\mu_{pp}(0,4)=0,75$$

$$\mu_{pg}(0,4)=0,45$$

2.1.3 AVALIAÇÃO DE REGRAS

O suporte ao processo de Avaliação de Regras recai sobre a Base de Conhecimento do Controlador Nebuloso. Seguindo a arquitetura de Sistemas Especialistas, a Base de Conhecimento é formada a partir de dois componentes: A Base de Dados (Memória de Trabalho) e a Base de Regras. (LEE ,1990 e BITTENCOURT,1998)

Base de Dados (memória de trabalho). A Base de Dados engloba a estruturação de dados dentro controlador, ou seja, como são apresentadas, armazenadas e processadas as variáveis no sistema. Fazem parte da estrutura

de dados, a discretização e normalização do Universo de Discurso (ganhos, intervalos e limites), as partições do espaço de entrada e saída do controlador e as representações das funções de pertinência. (LEE,1990 e JANTZEN,1998a)

Base de Regras. Uma das características principais de um Controlador Nebuloso é a sua estrutura de regras baseadas em conhecimento especialista. Estas regras usualmente tem a forma “SE-ENTÃO”. Segundo Jantzen (1998a) as regras podem conter múltiplas variáveis tanto no antecedente como no conseqüente. Desta forma, os controladores podem ser aplicados tanto em problemas MIMO (múltiplas-entradas-múltiplas-saídas) como também em SISO (Simples-Entrada-Simples-Saída). Entre algumas sintaxes de regras destacam-se:

- **Se** $X_1 = A_1$ **E** $X_n = A_n$ **ENTÃO** $Y=B$ (Mandani)
- **SE** $X_1 = A_1$ **E** $X_n = A_n$ **ENTÃO** $Y= p_0+p_1X_1+ \dots +p_nX_n$ (Sugeno)
- **SE** $X_1 = A_1$ **E** $X_n = A_n$ **ENTÃO** $Y= f(X_1, \dots ,X_n)$ (Outros)

A Figura (12) representa uma sintaxe de regra usual para Controladores Nebulosos, no caso, representa uma regra com duas condições no antecedente e uma saída no conseqüente.

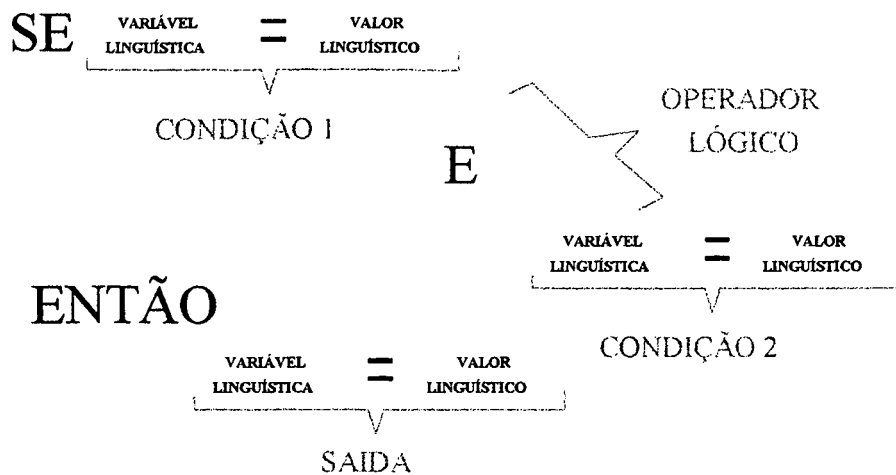


Figura 12 – Formato de uma regra constante de um bloco de avaliação de regras.

E a partir do número de variáveis lingüísticas envolvidas obtém-se a dimensão da tabela de regras a ser utilizada. Usualmente em um CLN-PD (Controlador Baseado em Lógica Nebulosa – Proporcional Derivativo) implementa-se uma tabela bidimensional, envolvendo o erro e a variação do erro, como variáveis lingüísticas principais. Sabe-se que quanto maior o número de variáveis lingüísticas condicionais das regras, maior a dimensão da matriz de regras, obedecendo a relação de um para um na dimensão da tabela.

Por exemplo:

- Três Variáveis Lingüísticas de entrada (erro, variação do erro e somatório do erro).
- 3-D para a matriz de representação de regras.

		Variação do erro					soma erro
		ng	nm	ze	pm	pg	
e r r o	ng	ng	nm	ze	pm	pg	
	nm	ng	nm	ze	pm	pg	
	ze	ng	nm	ze	pm	pg	
	pm	ng	nm	ze	pm	pg	
	pg	ng	nm	ze	pm	pg	
	pg	nm	ze	pm	pg	pg	
	pg	pm	nm	ze	pm	pg	pg

Figura 13 – Exemplo de tabela verdade para três variáveis lingüísticas.

Além da sintaxe padrão para as regras de um Controlador Nebuloso, muitas vezes é possível encontrar variáveis lingüísticas sobrecarregadas. Utilizando-se advérbios do tipo, muito, pouco, quase e etc.. pode-se sobrecarregar os Conjuntos Nebulosos existentes, transformando-os em outros mais específicos, sem que se demande grandes mudanças de estrutura do controlador. É necessário, no entanto, que o algoritmo do Motor de Inferência processe estes advérbios.

Na Tabela (1) pode-se encontrar um exemplo de tabela verdade bidimensional onde as variáveis lingüísticas são o erro, a variação do erro e a saída de controle. Cada Variável possui sete valores lingüísticos: ng – negativo grande ; nm – negativo médio ; ze – zero ; pm – positivo médio ; pg – positivo grande.

		var erro				
		ng	nm	ze	pm	pg
e r r o	ng	ng	nm	nm	pg	ng
	nm	pm	nm	ze	nm	ng
	ze	ze	ze	ze	ze	ze
	pm	nm	ze	pm	pg	pg
	pg	pg	pg	pg	pg	pg

Tabela 1 – Tabela Verdade.

Assumindo o CLN como um Sistema Especialista (SE), também é válida afirmação feita por BITTENCOURT (1998) que aquisição do conhecimento para a construção e manutenção da base de regras é uma das partes mais sensíveis do sistema. Como apresentado por LEE (1990) e NOBRE e PALHARES (1997) existem quatro formas de se adquirir conhecimento para a construção e manutenção da Base de Regras para um CLN:

- Baseando-se na experiência de um especialista ou no conhecimento da engenharia de controle.
- Observando as ações de controle de um operador.
- A partir de um modelo nebuloso do processo.
- Através de algoritmos de aprendizagem.

Motor de Inferência. A ativação das regras válidas, aplicação de operações lógicas sobre antecedentes e conseqüentes são atribuições do Motor de Inferência do CLN. Em sistemas que possuem algoritmo de aprendizado o Motor de Inferência também realiza a manutenção da base de conhecimento (MENDEL,1995). Operações foram descritas por ZADEH no seu artigo "FUZZY SET THEORY" (ZADEH,1965). Nas Figuras (14),(15), (16) e (17) são representadas graficamente as operações propostas por ZADEH (1965).

CONSIDERE OS SEGUINTE CONJUNTOS NEBULOSOS:

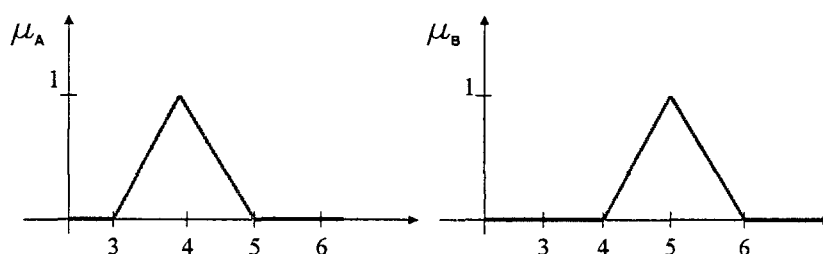


Figura 14 –Conjuntos Nedulosos

OPERAÇÃO E / \cap

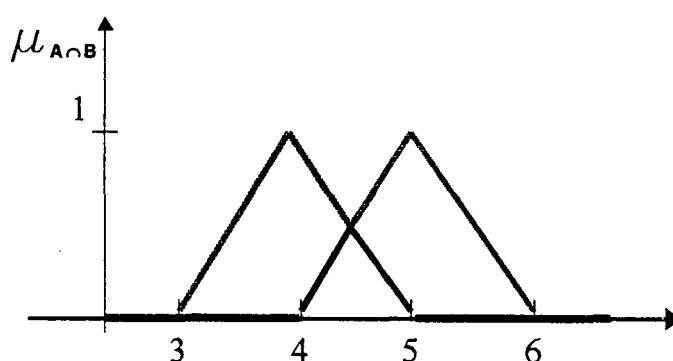


Figura 15 – Operação e / \cap sobre os conjuntos nebulosos da Figura(14).

OPERAÇÃO OU / \cup

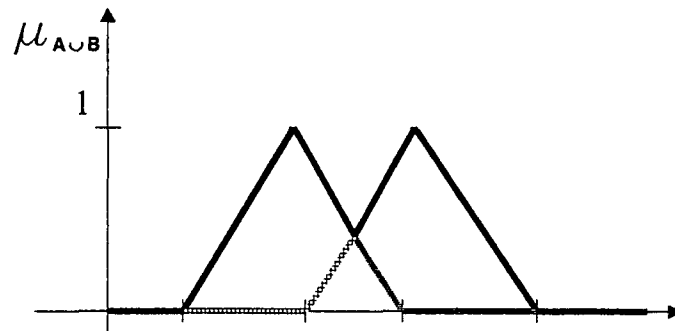


Figura 16 – Operação ou/ \cup sobre os conjuntos nebulosos da Figura(14).

OPERAÇÃO COMPLEMENTO / NEGAÇÃO

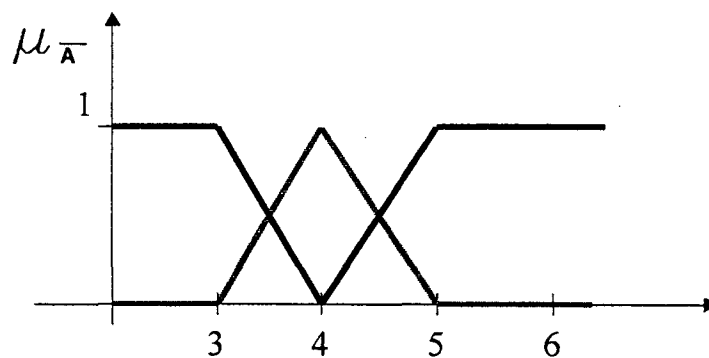


Figura 17 – Operação Complemento/negação sobre os conjuntos nebulosos da Figura (14).

Em notação matemática pode-se apresentar estas operações como:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Além do conceito de conjunto nebuloso, existe uma grande diferença entre a Lógica Nebulosa e a Lógica Clássica no âmbito das suas operações. Isto ocorre, devido ao fato de que as operações presentes lógica nebulosa estão sobre domínios contínuos e não apenas dicotômicos, característicos da lógica clássica.

As operações de união e a interseção, são definidas por uma família de operações chamadas de T-normas e T-conormas respectivamente, diversas implementações para estas regras podem ser encontradas na literatura. (BITTENCOURT,1998)

As operações apresentadas nas Figuras (15), (16) e (17) são as implementações mais usuais.

Sejam os Conjuntos Nebulosos apresentados na Figura (14) constituintes de duas variáveis lingüísticas distintas, ERRO (ERR) e VARIAÇÃO DO ERRO (VERR).

Onde,

$$\text{ERR} = 4,8$$

$$\text{VERR} = 4,5$$

Para a variável lingüística ERRO tem-se:

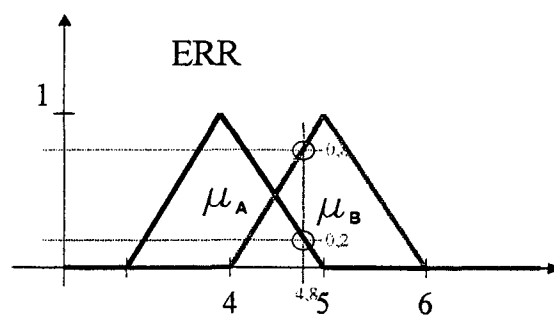


Figura 18 – Entrada para a variável ERR.

Para a variável lingüística VARIAÇÃO DO ERRO tem-se:

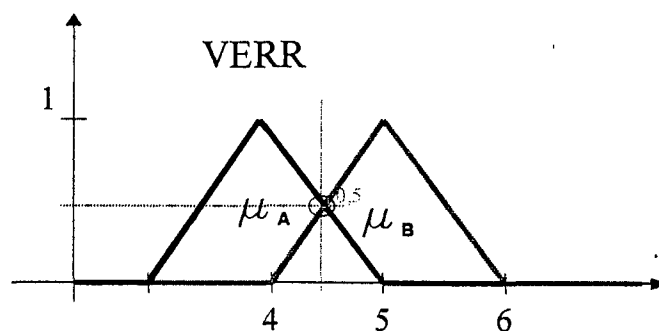


Figura 19 - Entrada para a variável VERR.

Deve ser avaliada a regra da Figura (20).

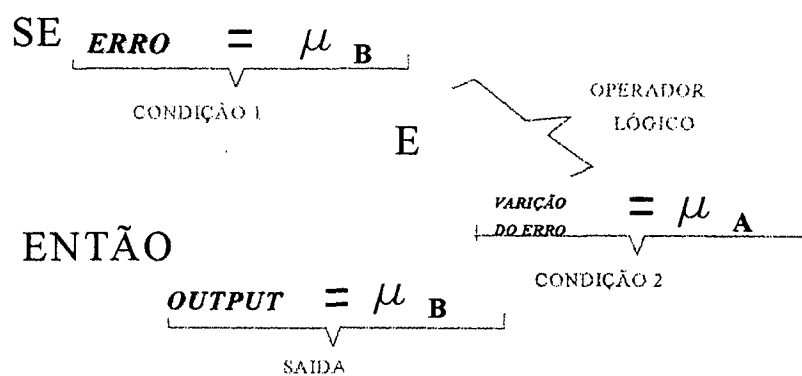


Figura 20 – Regra a ser avaliada.

Observar-se que cada condição construtiva da regra, possui um grau de pertinência associado a seu valor lógico. E para relacionar as duas condições utiliza-se o operador lógico “E” entre elas. Na resolução da regra, o valor resultante da operação lógica “E”, na forma de grau de adequação à regra, é transportado para conclusão. (JANTZEN, 1998a)

Na Figura (21) observa-se a representação gráfica da resolução da regra apresentada na Figura (20) para uma entrada de ERRO=4,8 e VARIAÇÃO DO ERRO=4,5.

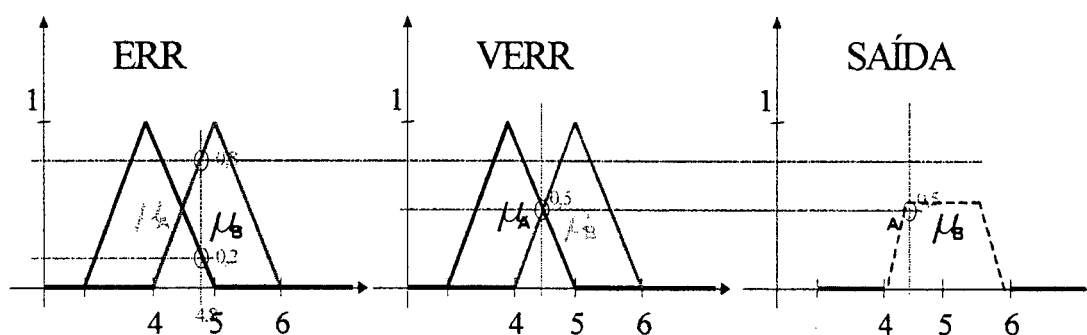


Figura 21 – Representação gráfica da avaliação de uma regra.

Durante um ciclo de controle, várias regras são resolvidas, devendo ser agregadas para compor a saída do processo de avaliação. De acordo com BITTENCOURT (1998), para agregar as conclusões das regras de um controlador baseado em lógica nebulosa em uma só saída podem ser utilizadas T-normas e T-conormas. No controlador implementado por Mandani (1974), é utilizada uma T-conorma “ou” para agrupar as conclusões das regras. A Figura (22) ilustra este processo.

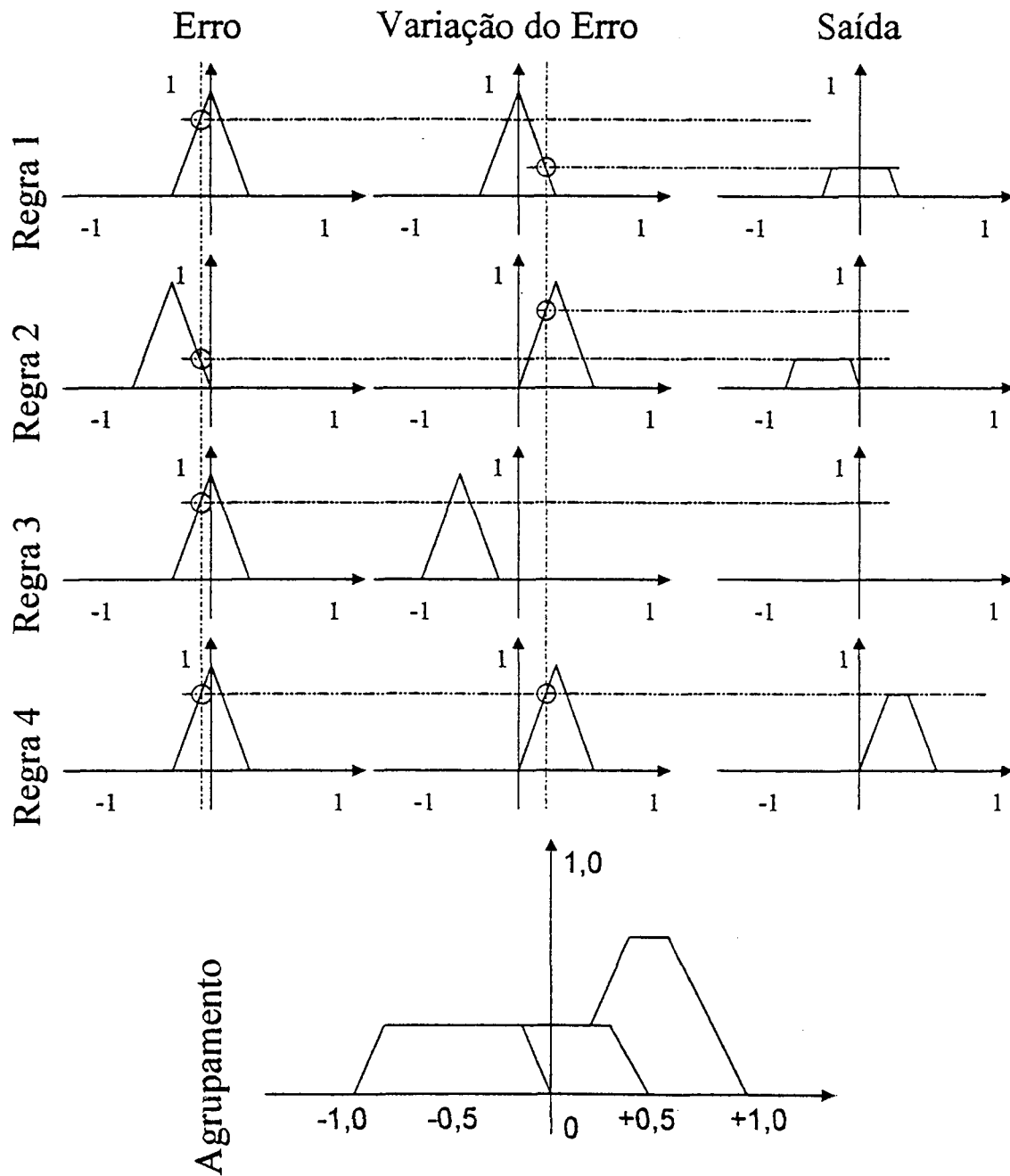


Figura 22 – Avaliação de um grupo de regras.

2.1.4 DESNEBULIZAÇÃO

Após a realização do processo de Avaliação de Regras sua saída é entregue ao processo de DESNEBULIZAÇÃO. Neste processo os resultados lingüísticos serão homogeneizados e resolvidos os conflitos para que possa ser gerada uma única saída numérica. A DESNEBULIZAÇÃO realiza o processo inverso da NEBULIZAÇÃO, transportando a entrada lingüística para a saída numérica (real), a qual irá atuar sobre o sistema que se deseja controlar. (MENDEL, 1995 e LEE, 1990)

Segundo MENDEL (1995), deve-se utilizar o critério de simplicidade computacional para a eleição do método de Nebulização e agregar a isto o objetivo que se deseja alcançar. Na literatura destacam-se cinco métodos de DESNEBULIZAÇÃO, são eles:

1. *Centro de Gravidade*. Este método poderia ser considerado como uma média ponderada dos elementos do conjunto suporte. A Equação (96) descreve matematicamente o método do centróide que pode ser visualizado na Figura (22).

$$u = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)} \quad (96)$$

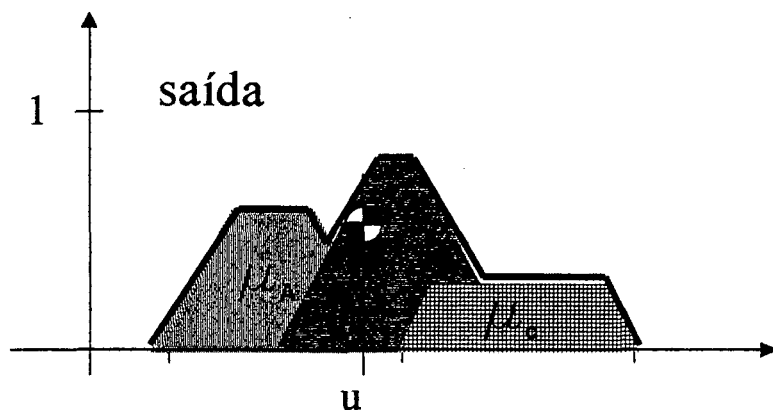


Figura 23 – Desnebulização utilizando o método do centróide.

2. *Maximo.* Neste método a abscissa do maior valor de pertinência para a Variável de Saída é definida como saída numérica. Este método pode tornar-se ambíguo, caso mais de um ponto tenha valor máximo. Desta forma este método possui alguma limitação e vai de encontro à habilidade de resolver conflitos característica da Lógica Nebulosa. A Figura (23) ilustra este método.

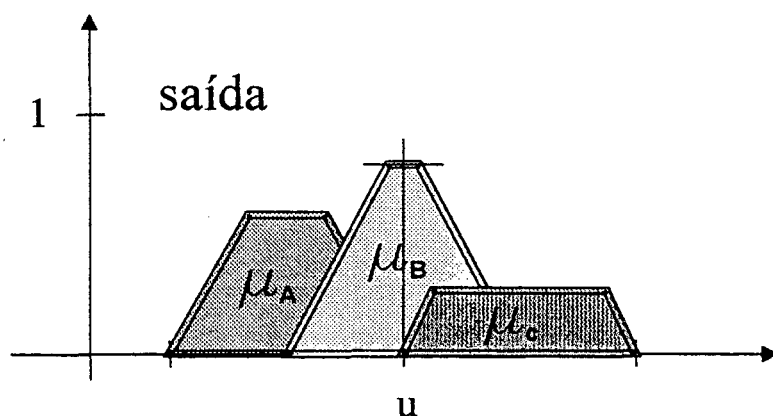


Figura 24 – Desnebulização utilizando o método do máximo.

3. *Média de Máximos.* Neste método uma média entre os pontos onde o grau de pertinência é máximo é feita. Este método é uma generalização do método de máximo, onde a sua limitação para situações onde possam haver mais de um ponto com valor máximo foi resolvida.

4. *Máximo extremo esquerdo / Máximo extremo direito.* É escolhida como saída a abscissa do ponto mais a esquerda ou a direita que possua máximo grau de pertinência. Segundo JANTZEN (1998a) este método é bastante utilizado em controle para robôs moveis a fim de evitar choques na sua trajetória, neste caso o controle deve se decidir entre duas direções e não concilia-las.
5. *Centro de Gravidade Ponderado.* Neste método, primeiro são calculados os centros de gravidade para cada Conjunto Nebuloso do conseqüente (\bar{x}_i) e em seguida calculado o centro de gravidade total. (ver Equação (97))

$$u = \frac{\sum_i \mu(\bar{x}_i) \bar{x}_i}{\sum_i \mu(\bar{x}_i)} \quad (97)$$

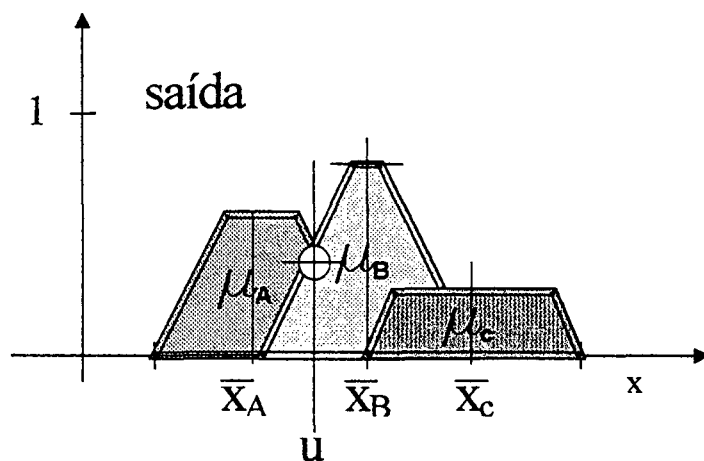


Figura 25 – Desnebulização utilizando o método do Centro de Gravidade Ponderado.

2.1.5 APRENDIZADO

Para a construção de um controlador nebuloso com propriedades de se auto-ajustar, é preciso que um algoritmo supervisor seja capaz de construir e

modificar a sua base de conhecimento. Foram identificados na literatura três estruturas básicas de algoritmo de aprendizado.

- Auto-organizavel. JANTZEN (1998b)
- Algoritmos Genéticos. HERRERA et al. (1995)
- Neuro-Nebuloso. (JANG e SUN, 1995)

Auto-ajustável. A idéia básica do controlador Nebuloso auto-ajustável é a utilização de duas tabelas, uma de tabela de regras e uma outra de desempenho. A cada ciclo de controle o algoritmo supervisor verifica se o controlador seguiu a tabela de desempenho, caso contrário uma penalidade é atribuída ao sinal de controle. Esta penalidade atua sobre a tabela de regras modificando o valor lingüístico do conseqüente da regra responsável por gerar o sinal de controle. (JANTZEN, 1998b)

Algoritmos Genéticos. Os algoritmos genéticos são considerados como um algoritmo de busca, baseados no comportamento dos genes na natureza. Neste processo de busca sobre o espaço de soluções de controle, as “populações” de configurações para o banco de conhecimento do controlador nebuloso são *cruzadas* e sofrem *mutações*. Como num processo de seleção natural, os *indivíduos* da *população* mais aptos para realizar o controle terão seus genes transmitidos as gerações seguintes de configurações de banco de conhecimento. Este processo ocorre até que as condições estabelecidas para o controle são atendidas. (HERRERA et al., 1995 e BITTENCOURT, 1998)

Neuro-Nebuloso. No controlador Neuro-Nebuloso a estrutura de um controlador nebuloso é transportada para uma estrutura similar na forma de redes neurais artificiais. Regras do tipo IF $X_1=A_1 \dots X_n=A_n$ THEN $Y=F(A_1, \dots, A_n)$ são utilizadas nesta implementação. Com isto, algoritmos de aprendizado específicos de redes neurais podem ser utilizados para a construção do banco de conhecimento do controlador. (JANG e SUN, 1995 e KOSKO, 1992)

2.2 CONSIDERAÇÕES

Após a descrição dos componentes básicos de um controlador baseado em lógica Nebulosa, resta discutir a sua implementação na forma de algoritmos e os aspectos práticos referentes a esta implementação. Sabe-se que diferentes arquiteturas de controle podem ser realizadas utilizando o Controlador Nebuloso, mas este tópico, junto com os aspectos da sua implementação, estão reservados para o capítulo seguinte desta dissertação.

Vale ressaltar que os controladores baseados em lógica Nebulosa é apenas uma, das diversas aplicações, da Lógica Nebulosa de ZADEH (1965) e sendo assim muitos aspectos pertinentes a formalização Lógica Nebulosa deixou-se de ser tratado neste capítulo. Existe ainda a formulação para implementação de algoritmos de aprendizado e uso de múltiplas variáveis de entrada e saída em controladores nebulosos que, como esta explicitado na introdução desta dissertação, não fazem parte do escopo deste trabalho.

CAPITULO 3

IMPLEMENTAÇÃO E RESULTADOS

Este capítulo estará dedicado à apresentação e discussão dos resultados alcançados durante o desenvolvimento deste trabalho. Em primeiro plano, será apresentado o sistema pneumático modelado, sua respectiva função de transferência. Em seguida serão discutidos os algoritmos utilizados para a construção do Controlador Nebuloso e sua estratégia de implementação.

Por fim a apresentação dos resultados das simulações e as conclusões que podem ser extraídas dos mesmos.

3.1 MODELO DO SERVO-POSICIONADOR PNEUMÁTICO

Utilizou-se um modelo de servo posicionador pneumático baseado no artigo de Tapio Virvalo (1989). Os dados de projeto, instalação e curvas experimentais do sistema pneumático podem ser encontrados nos trabalhos de dissertação de SANTOS(1996) e VIEIRA(1998). Desta forma, pôde-se ajustar o modelo matemático para que este representasse com fidelidade o sistema em estudo. Os desvios entre o modelo teórico e o real serão discutidos no momento de suas apresentações.

3.1.1 DADOS DOS SISTEMAS:

PISTÃO:

ÁREA A_1	0,00049 m ²
ÁREA A_2	0,00049 m ²
CURSO	1 m
MASSA	3,3 Kg
VOLUME MORTO	0,00002 m ³

SERVO

VÁLVULA

P_s	6 bar
ΔP	2 bar
Q_N	11,66 NI/s

Fonte: SANTOS(1996) e VIEIRA(1998)

3.1.2 FUNÇÃO TRANSFERÊNCIA:

Utilizando o modelo linear de terceira ordem, demonstrado no capítulo 2 desta dissertação, pôde-se definir a função de transferência para o sistema pneumático em questão. Esta modelagem forma o núcleo para que não-linearidades tipo saturações de sinais e zonas mortas possam ser incluídas. Na Figura(25) observa-se uma representação desta função de transferência.

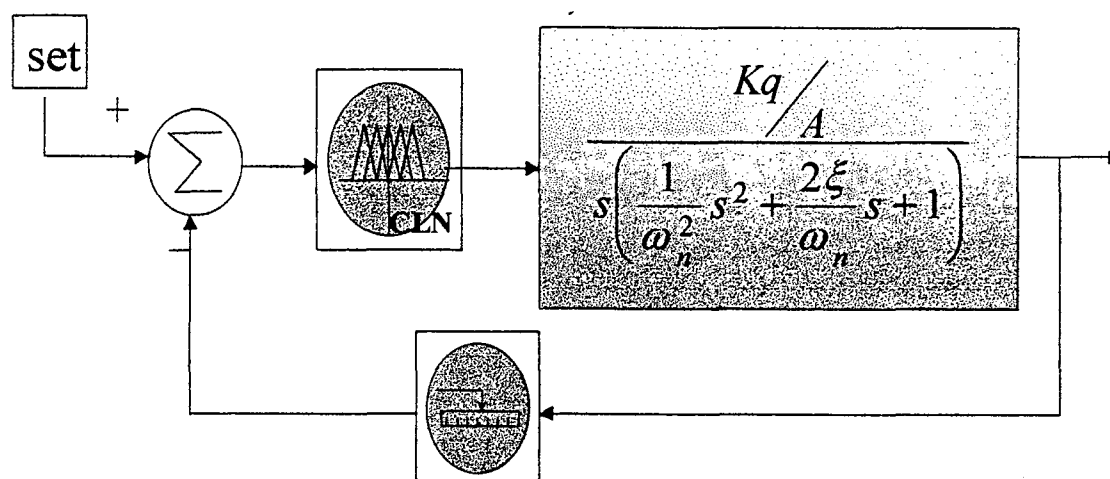


Figura 26 – Diagrama de Blocos de uma modelagem em terceira ordem de um sistema servo pneumático controlado por um CLN.

SANTOS

β	KP	4,2bar
ω_n	$\sqrt{\frac{4 \beta A^2}{V_t M}}$	15,20 rad/s
ξ	$f \omega_n / 2K_{pneu}$	0,46
K_q	$\frac{Q_N \sqrt{1/3 P_S}}{\left(\frac{2}{3} P_S + 1\right) X_1}$	0,629l/s

Para efeito de linearização do modelo baseado nos dados apresentados por SANTOS (1996) em seu trabalho de dissertação, foi utilizada a medição da força de atrito, ali apresentada, para calcular o coeficiente de amortecimento do sistema. Utilizando-se os valores máximos de atrito e velocidade atingidos obtém-se:

$$f = F_{at}/V_{max} \therefore f = 180/3$$

$$f = 60 \text{ Ns/m}$$

Esta aproximação mostrou-se satisfatória para representação do sistema em questão, pois comparando os resultados de SANTOS (1996) e VIEIRA (1998) para o sistema em malha aberta, tanto o seu modelo teórico não-linear quanto as curvas obtidas experimentalmente, o modelo aqui proposto não apresentou desvios significativos. Considerou-se válida a afirmativa de VIRVALO (1989), que um modelo linear de terceira ordem é adequado para o estudo de controle em servo mecanismos. O próprio VIEIRA (1998) chegou a esta conclusão após verificações experimentais, mas alertando para a necessidade de refinamento do modelo.

3.2 CURVAS EM MALHA ABERTA

Utilizando-se dos dados dimensionais e experimentais coletados por SANTOS (1996) para desenvolvimento da sua dissertação de mestrado, montou-se o modelo linear de terceira ordem para o sistema. Para efeito de melhor comparação, foram realizadas simulações com diferentes referências em dregrau de abertura de válvula de controle. Ver Figuras (27), (28), (29) e (30).

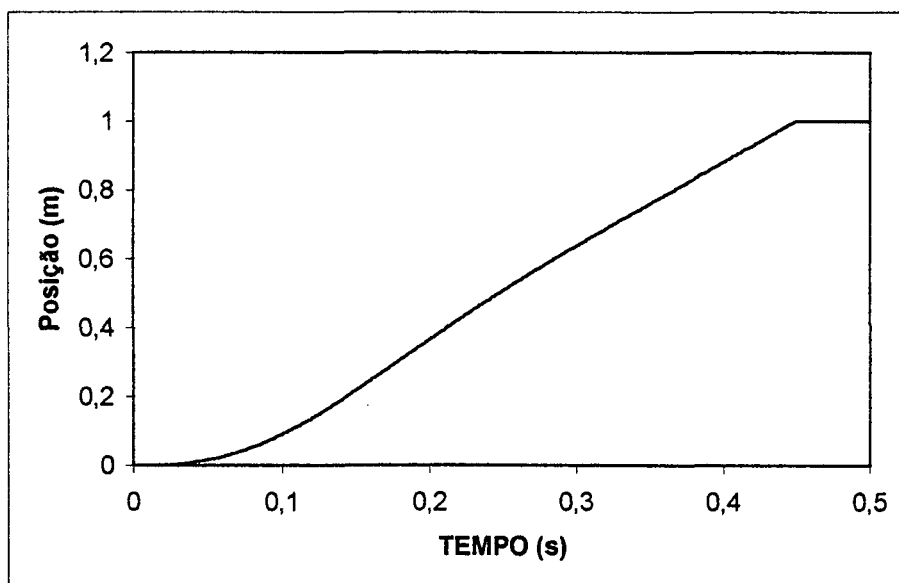


Figura 27 - Curva em malha aberta do modelo linear de terceira ordem com 50% de abertura da válvula (simulação). (SANTOS,1996)

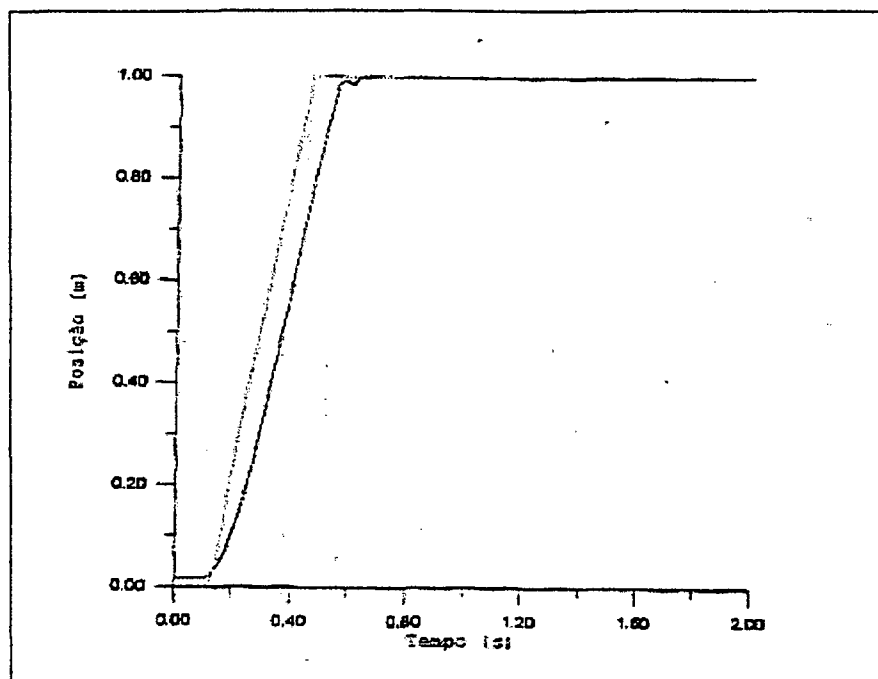


Figura 28 – Curva em malha aberta experimental e teórica do modelo não-linear com 50% de abertura. (linhas pontilhadas – modelo não-linear ; linha cheia – resultado experimental) (SANTOS,1996)

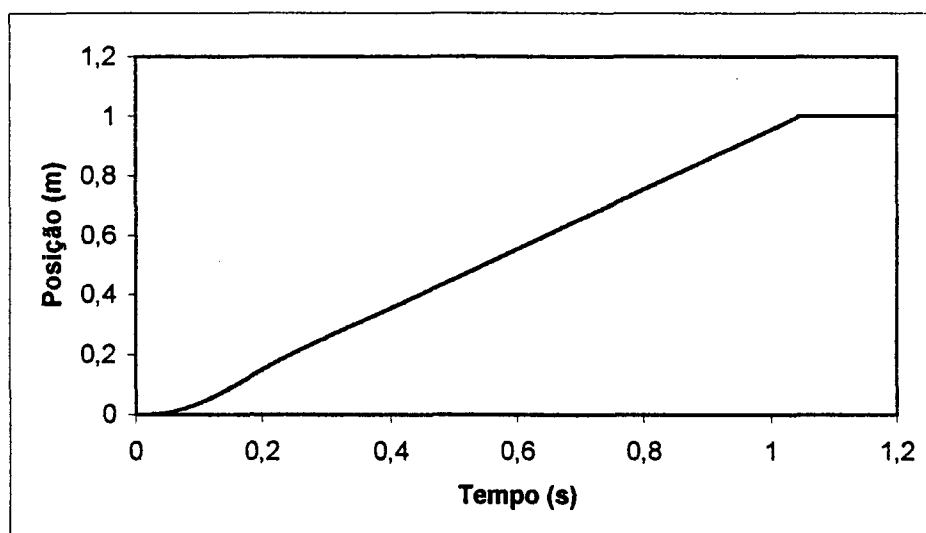


Figura 29 - Curva em malha aberta do modelo linear de terceira ordem com 20% de abertura da válvula (simulação). (SANTOS,1996)

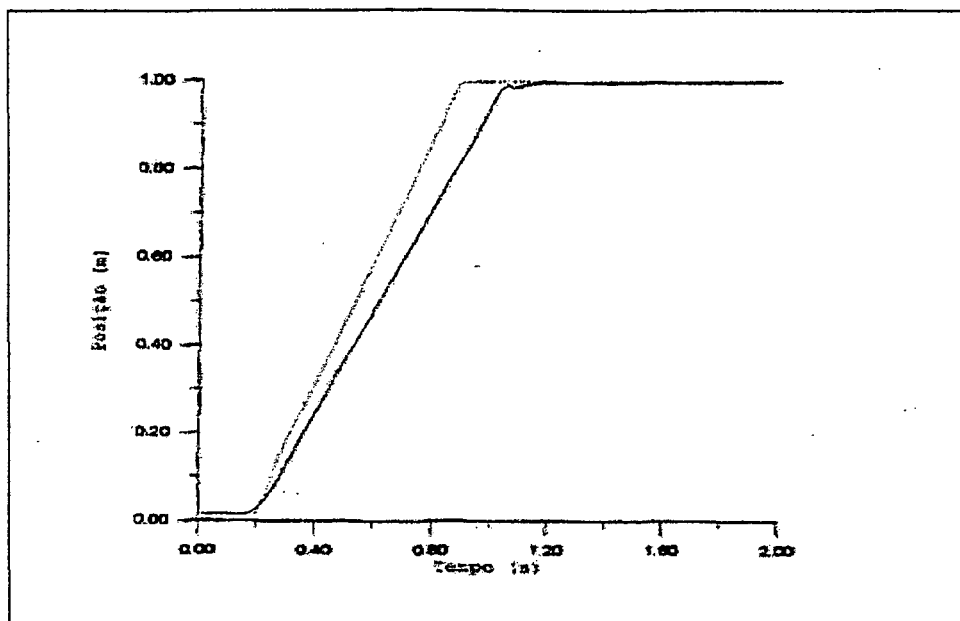


Figura 30 - Curva em malha aberta experimental e teórica do modelo não-linear com 20% de abertura. (linhas pontilhadas – modelo não-linear ; linha cheia – resultado experimental) (SANTOS,1996)

3.3 ALGORÍTMOS DO CONTROLADOR FUZZY

A linguagem de programação e a plataforma onde o sistema será executado, são pontos essenciais para a construção do Controlador Nebuloso. A estes estão vinculados o desempenho controlador, pois a sua velocidade de execução e sua flexibilidade em estrutura de dados são dependentes destes elementos.

3.3.1 A LINGUAGEM DE PROGRAMAÇÃO

A construção do controlador foi distribuída em três etapas. A primeira tratou do desenvolvimento e seleção de diferentes algoritmos para o processo de Nebulização. Buscou-se na literatura algoritmos para a Nebulização consolidados e também optou-se para o desenvolvimento de um algoritmo próprio. A Segunda etapa foi caracterizada por testes de velocidade e efetividade dos algoritmos escolhidos, para que fosse definida a melhor solução para a aplicação. Na terceira etapa foram escolhidos os algoritmos dos processos de avaliação de regras e Desnebulização e a compilação do controlador propriamente dito. Vale ressaltar que para os algoritmos de Avaliação de Regras e Desnebulização não foram feitos testes de desempenho.

Ao tempo em que se estava procurando uma linguagem de programação que se adequasse a aplicação, foram estabelecidos critérios para esta escolha, já que as particularidades de uma aplicação em controle exigem características peculiares para a linguagem. Foi estabelecido que a linguagem em questão deveria aliar flexibilidade, com velocidade e natureza científica. Foi cogitada a utilização do FORTRAN para a implementação mas, optou-se por descartá-la devido a sua rigidez de compilação e baixa estruturação de dados.

Uma outra linguagem especializada, em Inteligência Computacional (ex. LISP, PROLOG e etc.), poderia ser uma alternativa passível de utilização, mas a

inexistência de compiladores comerciais e sua falta de portabilidade para a plataforma Windows® contribuíram para a desistência de sua utilização.

Por fim, foi escolhida a linguagem de programação C/C++ para a implementação dos algoritmos do controlador, pois esta, constitui-se numa das poucas linguagens que reúne em si, todas as necessidades para esta implementação. A flexibilidade do C/C++ proporcionou um ganho produtivo imenso a este trabalho, já que a simulação pode ser realizada através da compilação dos algoritmos do controlador em uma .DLL para Windows® e a sua posterior inserção no Software de simulação VISSIM®.

O Software de simulação VISSIM® foi utilizado devido a sua fácil manipulação, a utilização blocos de não-linearidades e a possibilidade de utilização de DLLs como blocos pertencentes ao sistema. Devido a sua característica, de permitir a utilização de blocos DLL, foi possível inserir o bloco do controlador Nebuloso na simulação.

3.3.2 ALGORITIMOS

Para aplicações simples (ex.: duas variáveis) em processos onde as velocidades de resposta sejam baixas, aliado ao uso dos atuais microcomputadores, o tempo de execução do controlador não é problema. No entanto, quando são realizadas aplicações multivariáveis em processos onde a velocidade de resposta do controlador pode ser decisiva entre a estabilidade e

a instabilidade o tempo de execução do controlador é um parâmetro fundamental. Desta forma, neste trabalho de dissertação decidiu-se por otimizar o processo da nebulização através de um algoritmo mais eficiente. É possível que a nebulização não seja o processo mais lento em um CLN, mas devido ao fato da execução do controlador ser feita em forma seqüencial, ganhos de velocidade em qualquer dos processos será uma ganho global.

Visando a implementação do processo de Nebulização, foram avaliados quatro algoritmos. Dois deles foram retirados da bibliografia especializada e um outro foi desenvolvido visando a avaliação de uma nova estratégia para a realização da FUZZYFICAÇÃO, uma adaptação do conceito de Meta-regras (BARCZAK e ROSA, 1993).

O primeiro algoritmo implementado foi a Fuzzyficação Direta Simples (FDS), assim nomeado devido a sua característica de utilização de variáveis locais para a representação de cada uma das variáveis lingüísticas. Esta estrutura de dados provoca um enorme consumo de memória e um elevado número de instruções e, sendo assim, conseqüentemente, possuindo baixas velocidades de acesso e grande tamanho de código compilado (LISTAGEM 1).

Uma progressão sobre o algoritmo FDS é Fuzzyficação Direta Simples com Vetores (FDSV), utilizando-se uma estrutura de dados através da indexação de vetores. Economizou-se, com isto, linhas de código para elaboração do algoritmo. Esta é uma simplificação do algoritmo proposto por COX (1994). (LISTAGEM 2).

Avançando um pouco mais nos conceitos de estruturação de dados, estaria a Fuzzyficação Direta com Listas (FDL), que trata da introdução do uso de listas encadeadas para armazenamento e manipulação dos dados através de alocação dinâmica. Este algoritmo foi adaptado do artigo de G. VIOT (1993). Esta estratégia de alocação de valores possui características muito peculiares como a facilidade de localização e manipulação de dados, o que possibilitou ganhos de velocidades significativos para o controlador.

A estrutura embrionária para o desenvolvimento da lista encadeada foi a seguinte:

```
struct fp_tipo    {  
  
    char nome [10];  
  
    float gp;  
  
    float pt_1;  
  
    float pt_2;  
  
    struct fp_tipo *next;  
  
};
```

A partir da criação desta lista, os dados seriam acessados e manipulados para a realização da FUZZYFICAÇÃO (LISTAGEM 3).

Foi adaptado o conceito de Meta Regras (BARCZAK, 1993) para construção de um algoritmo baseado numa árvore decisória ternária. Desta forma desejava-se que fossem maximizadas as operações lógicas e minimizadas as operações numéricas, revertendo o tempo otimizado em ganho na velocidade de execução do algoritmo.

Esta árvore decisória seleciona os cálculos que devem ser executados, direcionando os esforços de processamento e obtendo-se maior efetividade na execução do algoritmo. Um exemplo da árvore decisória pode ser observada na Figura (30).

Esta estratégia apenas realiza cálculos matemáticos na extremidade da árvore, minimizando a quantidade de cálculos necessários a Nebulização (LISTAGEM 4).

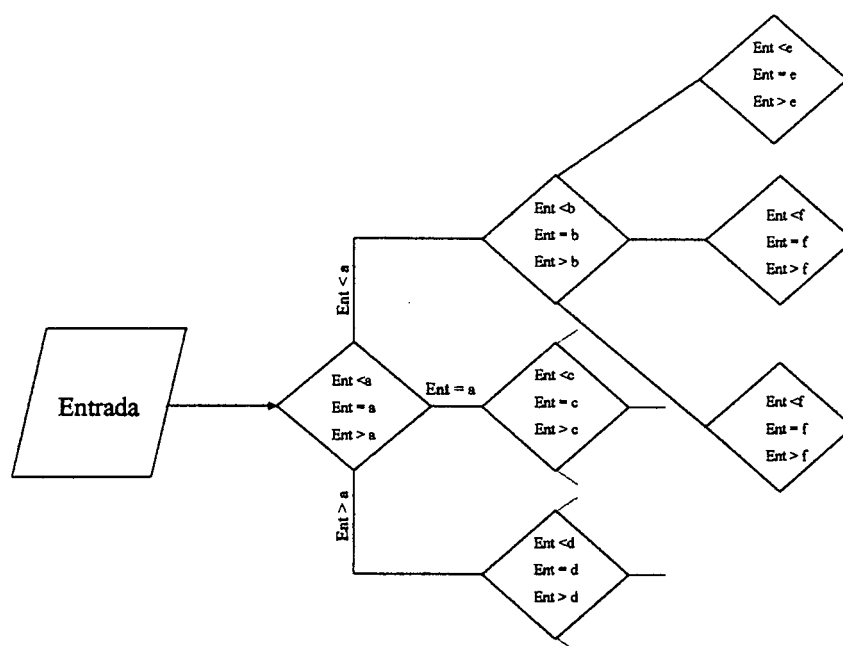


Figura 31 – Árvore ternária.

3.3.3 AVALIAÇÃO

Foram estabelecidos dois critérios para a avaliação da performance dos algoritmos. Em primeiro lugar a velocidade de execução e, em segundo, o espaço ocupado em disco pelo código compilado.

Para a medição da velocidade de execução dos algoritmos, foi realizada uma montagem em laboratório utilizando-se a porta paralela (lpt1) de um PC-486 DX66 e um osciloscópio . Foi realizada uma varredura em cada um dos algoritmos variando-se os valores de entrada e pondo o algoritmo em ciclo contínuo, para que o seu período de execução fosse medido.

Constatou-se que a diferença de velocidade de execução do algoritmo baseado em Meta Regras para a Nebulização é muito significativa. Seu ganho de velocidade chega a ordem de 7 (sete) vezes, se comparado com o algoritmo mais lento. É válido salientar, também, que, sua pequena flutuação de velocidade, relacionada com os valores de entrada para Nebulização, torna-se de uma característica muito importante para a homogeneização da ação do controlador dentro do Universo de Discussão. O resultado desta medição pode-se ser avaliado no gráfico da Figura (31).

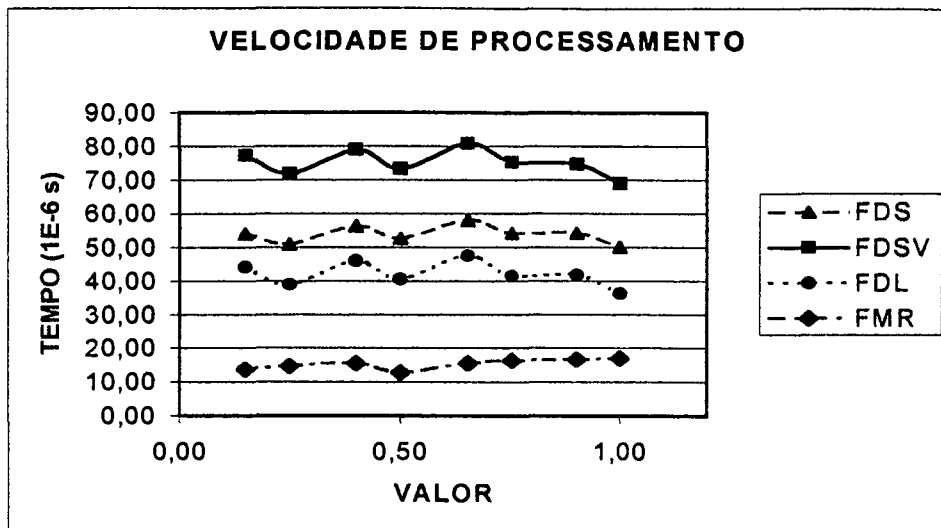


Figura 32 – TEMPO x VALOR FUZZYFICADO.

O segundo critério para avaliação dos algoritmos foi o espaço em disco ocupado pelo algoritmo compilado, já que, para controladores dedicados, a memória disponível para armazenagem destes programas é limitada.

Constatou-se que não existe uma relação direta entre espaço em disco ocupado e velocidade de processamento. A implementação, utilizando a derivação do conceito de Meta-Regras trata-se de uma estratégia que preserva a melhor relação entre espaço alocado e velocidade..



Figura 33 –Espaço alocado em disco para os diferentes algoritmos de Nebulização.

Finalizada a avaliação dos algoritmos, optou-se pela implementação do algoritmo FMR, tendo em vista o seu desempenho. Para a implementação dos demais processos do controlador (Avaliação de Regras e Desnebulização), foram utilizados algoritmos propostos por VIOT(1993).

A avaliação de regras teve como núcleo laços “FOR” aninhados varrendo uma lista encadeada para frente, com a seguinte estrutura:

```
struct regras {
    char if1[3];
    char if2[3];
    char then[3];
    struct regras *ptProxima;
};
```

O algoritmo implementado para a desnebulização foi *Centro de Gravidade Ponderado*.

3.4 CONTROLADORES E CURVAS EM MALHA FECHADA

O debate sobre o uso de diferentes técnicas controle e dentre elas o controlador Nebuloso, tem motivado muito trabalhos e publicações, como : LI e LAU (1989); COLEMAN e GODBOLE(1994) e COELHO et al. (1997). Nestes artigos é unanime a opinião de que os controladores baseados em lógica nebulosa apresentam-se suficientemente robustos para variações de sinal e perturbações de carga.

Devido a sua baixa rigidez o sistema servopneumático é sensível as variações de carga. Em um aplicação real de posicionamento (ex.: Braço mecânico) onde uma carga seria transferida de um ponto a outro, quando o posicionador deixasse a carga poderia ocorrer uma degradação do seu desempenho ou numa situação mais crítica tornar-se instável.

Definiu-se cinco estratégias de controle para serem implementadas e avaliadas através de critérios de desempenho em duas situações de carga. Uma seqüência de sinais foi criada para submeter o sistema a três tipos de sinal , ao degrau, a rampa de 1ª ordem e a rampa de 2ª ordem. Como critérios de avaliação foram estabelecidos os seguintes parâmetros:

- *Rise time*
- *Overshot*

- Somatório Absoluto do Erro

$$(\sum |e| = \sum_{k=1}^n |e(k)|) \quad (98)$$

- Variância da ação de controle

$$(J = \frac{1}{N} \sum_{k=1}^N [u(k) - \bar{u}]^2) \quad (99)$$

As seguintes arquiteturas de controle foram implementadas para avaliação:

- i. Controlador Nebuloso – Proporcional Derivativo (CLN-PD)
- ii. Proporcional Derivativo Integral – clássico-
- iii. Controlador Nebuloso Supervisório – direto
- iv. Controlador Nebuloso Supervisório – inverso
- v. Controlador Nebuloso – Proporcional Integral + Proporcional Derivativo - clássico

3.4.1 Controlador Nebuloso – Proporcional Derivativo

Em um controlador baseado em Lógica Nebuloso a sua sintonia é realizada através da construção da sua tabela verdade. Muitas vezes esta tabela é concebida a partir da codificação manual da experiência preexistente de um controlador humano. Ou automaticamente, através de ensaio sobre o sistema simulado ou real, utilizando como ferramenta algoritmos de aprendizado. (LEE,1990 e JANTSEN,1998)

Propostas de procedimentos para a ajuste manual destes controladores já foram feitas, muitos deles baseados em similaridades com a teoria de controle clássica. Mas existe um ponto em comum que é a necessidade de um conhecimento comportamento dinâmico do sistema controlado para a execução de um ajuste efetivo. (TAN, 1994 e NOBRE e PALHARES, 1997)

A arquitetura escolhida para o controlador Nebuloso pode ser observada na Figura (33). A opção por um Controlador Nebuloso Derivativo estabelece o uso do erro e da variação do erro como Variáveis lingüísticas. Como recomendado por JANTZEN (1998a), utilizou-se as formas triangulares para as funções de pertinência (ver Figura (34)) para um Universo de Discurso normalizado em $[-1,1]$ na partição das variáveis lingüísticas. A definição dos demais parâmetros foram a partir de tentativa e erro.

Para este controlador foram necessários definir:

- Variáveis Lingüísticas
- Universo de Discurso
- Funções de Pertinência
- Número Valores Lingüísticos
- Ganhos do Erro e Variação do Erro
- Ganho da Saída
- Base de Regras (ver Tabela (2))

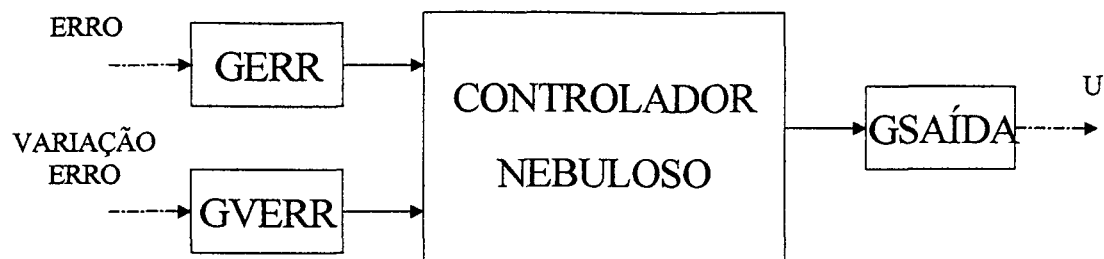


Figura 34 – Arquitetura do CLN-PD.

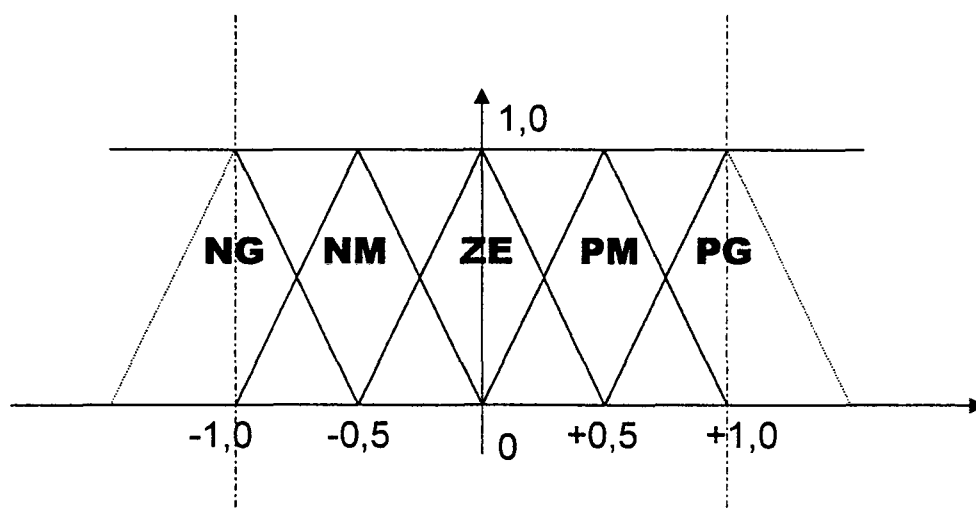


Figura 35 – Partição das Variáveis Linguísticas.

var erro

	ng	nm	ze	pm	pg
ng	ng	nm	nm	pg	ng
nm	pm	nm	ze	nm	ng
ze	ze	ze	ze	ze	ze
pm	nm	ze	pm	pg	pg
pg	pg	pg	pg	pg	pg

e
r
r
o

Tabela 2 – Base de Regras para CLN-PD.

Ganhos do Controlador:

- Ganho da Variável *Erro* (GERR): 1,8
- Ganho da Variável *Varição do Erro* (GVERR): 900
- Ganho da Saída (GSAÍDA): 1,2

A superfície de controle gerada por este banco de conhecimento pode ser vista na Figura (35).

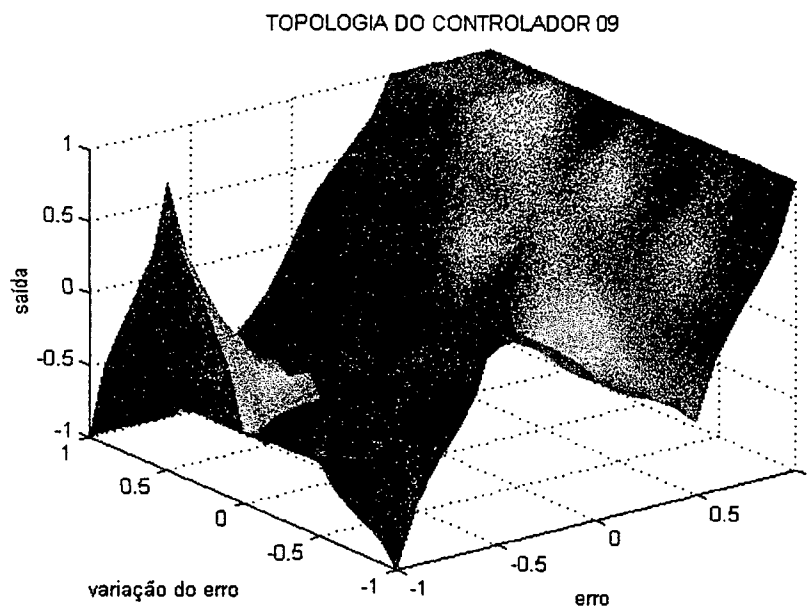


Figura 36 – Topologia da Superfície da Saída do CLN-PD.

As respostas em malha fechada para este controlador estão apresentadas na seguinte disposição:

	Massa 3,3 kg	Massa 0,5 kg
POSIÇÃO	Figura (36)	Figura(38)
ESFORÇO DE CONTROLE	Figura (37)	Figura (39)

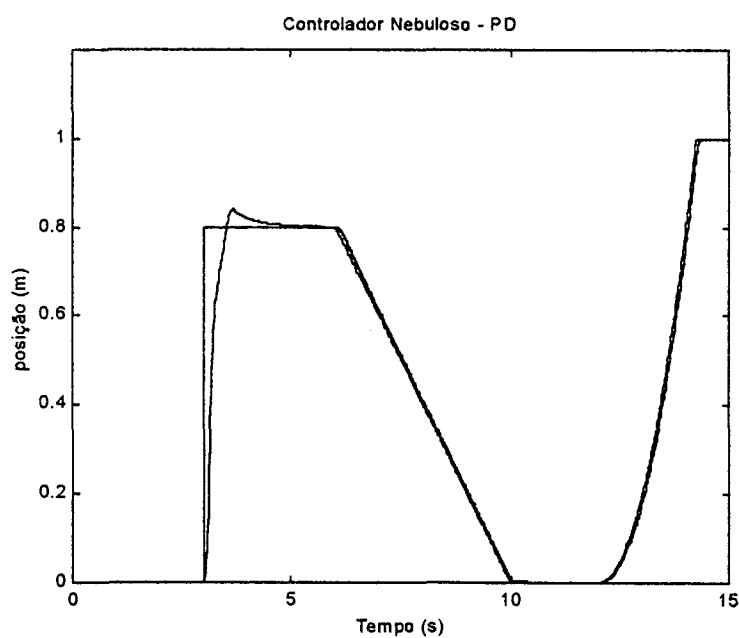


Figura 37 – Controlador Lógico Nebuloso – PD (Posição)
MASSA: 3,3 kg

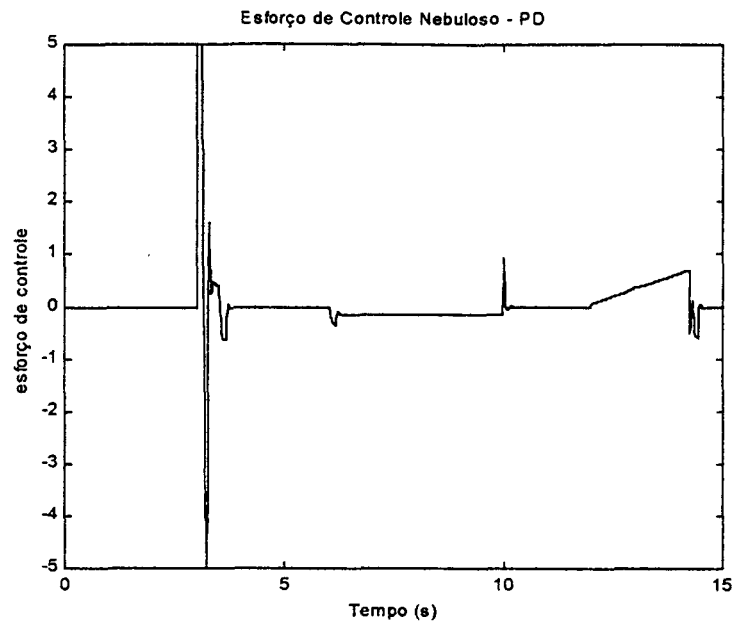


Figura 38 – Controlador Lógico Nebuloso – PD (Esforço de controle)
MASSA: 3,3 kg

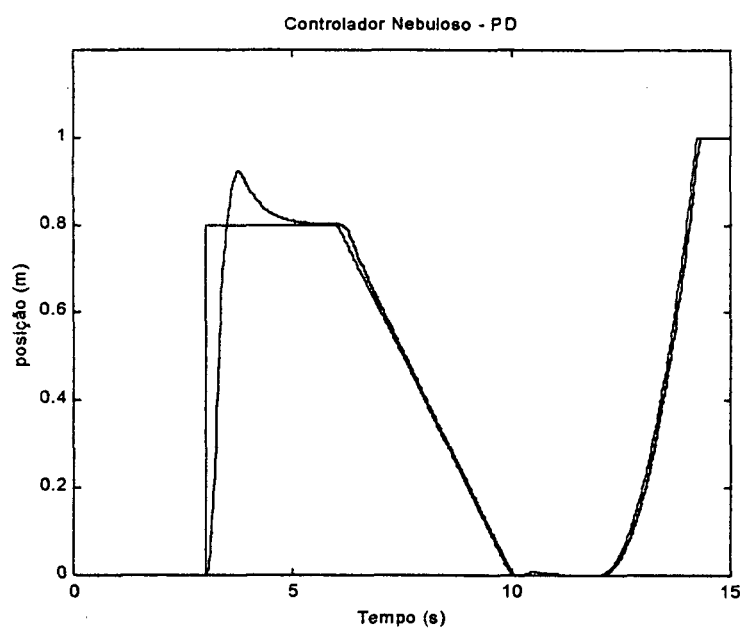


Figura 39 – Controlador Lógico Nebuloso – PD (Posição)
MASSA: 0,5 kg

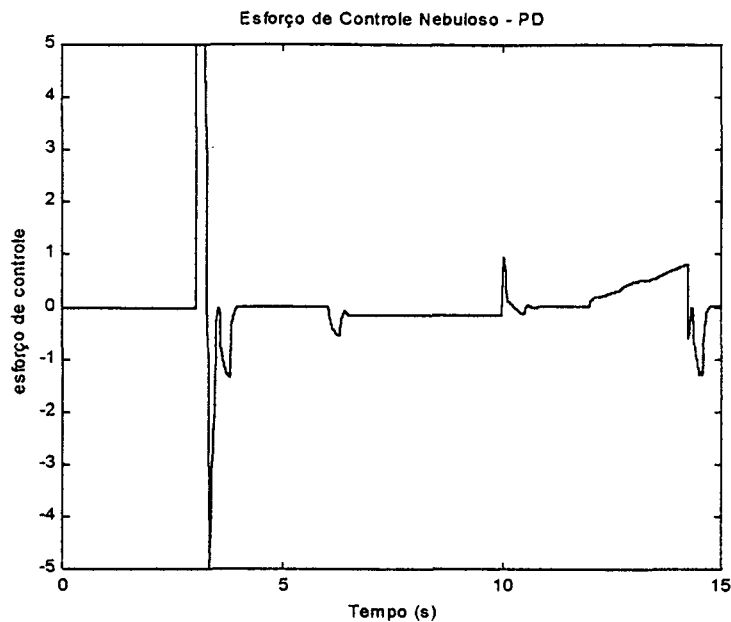


Figura 40 – Controlador Lógico Nebuloso – PD (Esforço de controle)
MASSA: 0,5 kg

3.4.2 Controlador Proporcional Integral e Derivativo

O controlador PID foi escolhido devido a sua simplicidade de aplicação, suas características dinâmicas e robustez. Além disto, por ser um das técnicas de controle mais difundida na indústria (COLEMAN e GODBOLE, 1994 e COELHO e COELHO, 1999). No entanto, segundo COELHO e COELHO (1999), PID's lineares convencionais podem apresentar dificuldades em lidar com sistemas dinâmico complexos, grandes atrasos e não-linearidades.

Definiu-se um PID linear convencional do tipo:

$$\frac{U(s)}{E(s)} = K \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (100)$$

Os ganhos das bandas proporcional, integral e derivativa foram ajustados usando o método de tentativa e erro (JANTZEN, 1998c e FRANKLIN et al., 1994):

- $K=1,2$
- $T_D = 0,08$
- $1/T_I = 0,001$

Apesar da planta em questão já possuir um integrador, foi ajustado um pequeno ganho integral para garantir o erro nulo em regime permanente. Sabendo-se que na planta real o integrador da função transferência não se comporta como no modelo linear. Para evitar a amplificação de ruídos numa aplicação real, recomenda-se a utilização de uma realimentação *lead* no lugar da derivativa. (FRANKLIN et al., 1994)

As respostas em malha fechada para este controlador estão apresentadas na seguinte disposição:

	Massa 3,3 kg	Massa 0,5 kg
POSIÇÃO	Figura (40)	Figura(42)
ESFORÇO DE CONTROLE	Figura (41)	Figura (43)

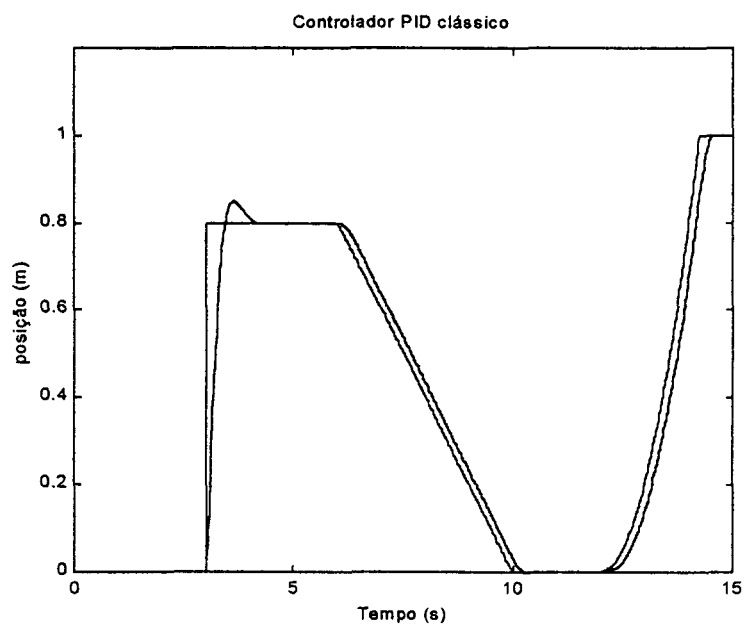


Figura 41 – PID clássico (Posição)

MASSA: 3,3 kg

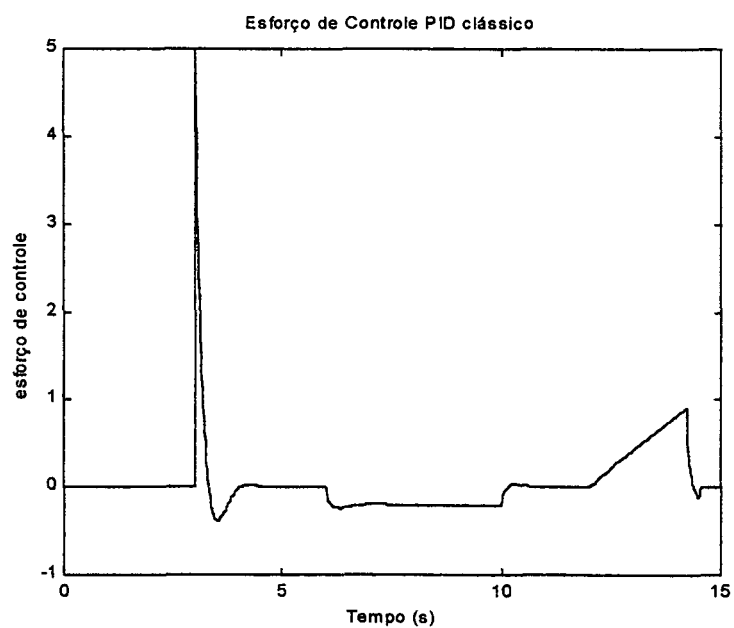


Figura 42 – PID clássico (Esforço de controle)

MASSA: 3,3 kg

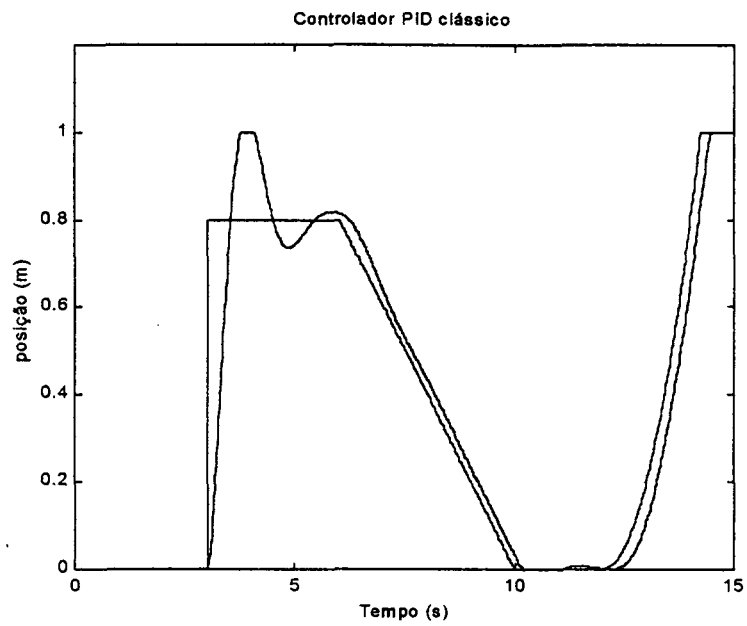


Figura 43 – PID clássico (Posição)

MASSA: 0,5 kg

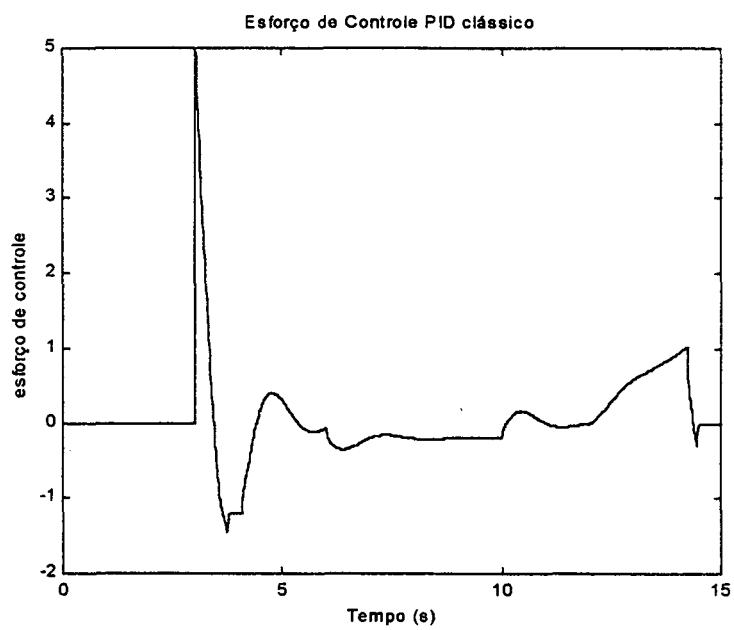


Figura 44 – PID clássico (Esforço de controle)

MASSA: 0,5 kg

3.4.3 Controlador PID + Nebuloso Supervisório (Direto)

Atualmente, encontra-se na literatura arquiteturas de controladores nebulosos realizando tarefas supervisórias sobre controladores tradicionais (ex. PID). (ALTROCK, 1999 e YAZDI, 1997)

Nesta arquitetura propõe-se um controlador nebuloso ajustando o ganho proporcional do controlador PID (Figura (44)). Realizando uma ação direta sobre o ganho. O ganho é crescente a medida que o erro vai diminuindo (ver Figura 47).

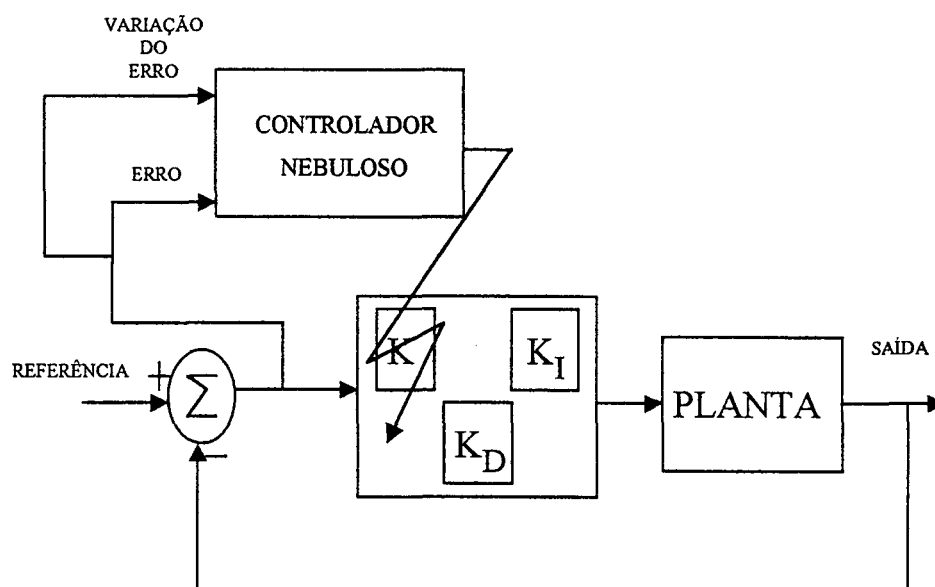


Figura 45 – Arquitetura do controlador PID+CLN supervisório

O Banco de Memória deste controlador é igual ao do item 3.4.1.

		var erro				
		ng	nm	ze	pm	pg
e r r o	ng	ng	ng	ng	ng	ng
	nm	ng	ng	ng	ng	ng
	ze	pm	pm	ze	ze	ze
	pm	pg	pg	pg	pg	pg
	pg	pg	pg	pg	pg	pg

Tabela 3 – Base de Regras para PID+CLN supervisorio.

Ganhos do Controlador:

- Ganho da Variável *Erro* (GERR): 1,5
- Ganho da Variável *Variação do Erro* (GVERR): 1200
- Ganho da Saída (GSAÍDA): 1,2

Os ganhos das bandas integral e derivativa foram ajustados usando o método de tentativa e erro (JANTZEN, 1998c e FRANKLIN et al., 1994):

- $T_D = 0,20$
- $1/T_I = 0,001$

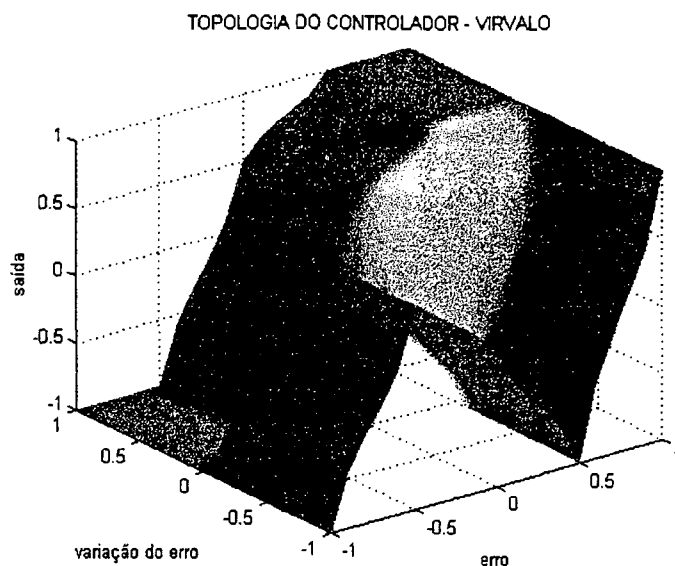


Figura 46- Topologia da Saída do CLN supervisorio direto

As respostas em malha fechada para este controlador estão apresentadas na seguinte disposição:

	Massa 3,3 kg	Massa 0,5 kg
POSIÇÃO	Figura (46)	Figura(49)
ESFORÇO DE CONTROLE	Figura (47)	Figura (50)
VARIAÇÃO DO GANHO	Figura (48)	Figura (51)

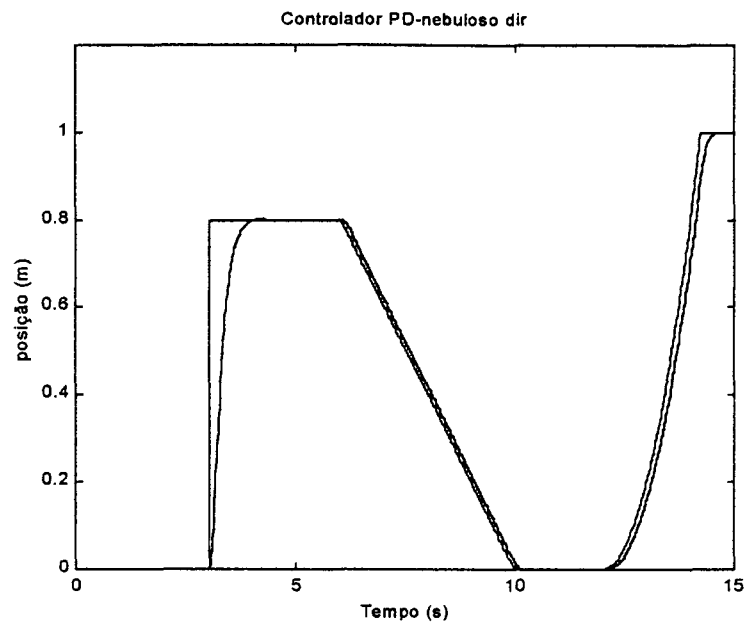


Figura 47 – PID+CLN supervisorio direto (Posição)
MASSA: 3,3 kg

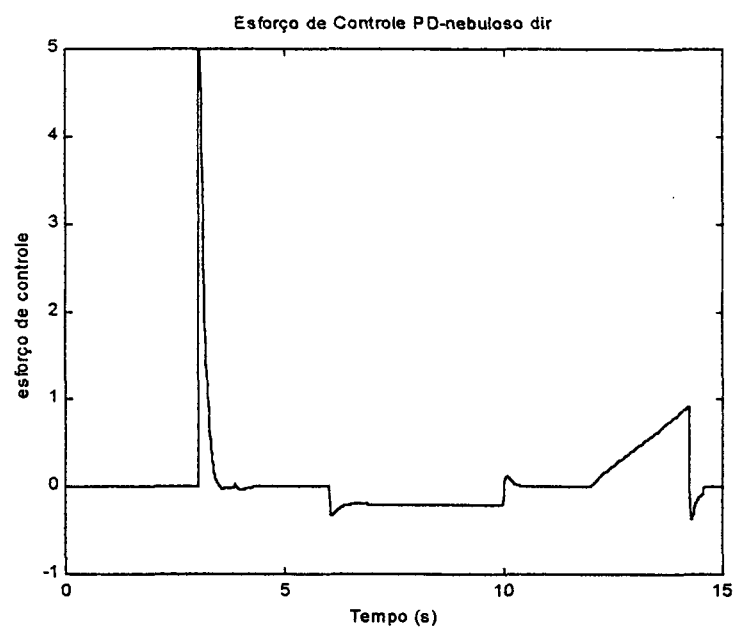


Figura 48 – PID+CLN supervisorio direto (Esforço de controle)
MASSA: 3,3 kg

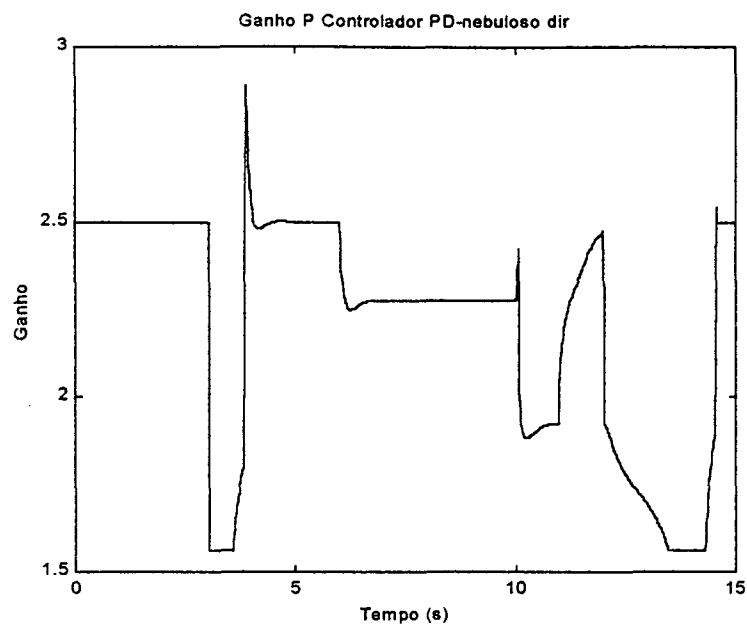


Figura 49 – PID+CLN supervisório direto (variação do ganho)
MASSA: 3,3 kg

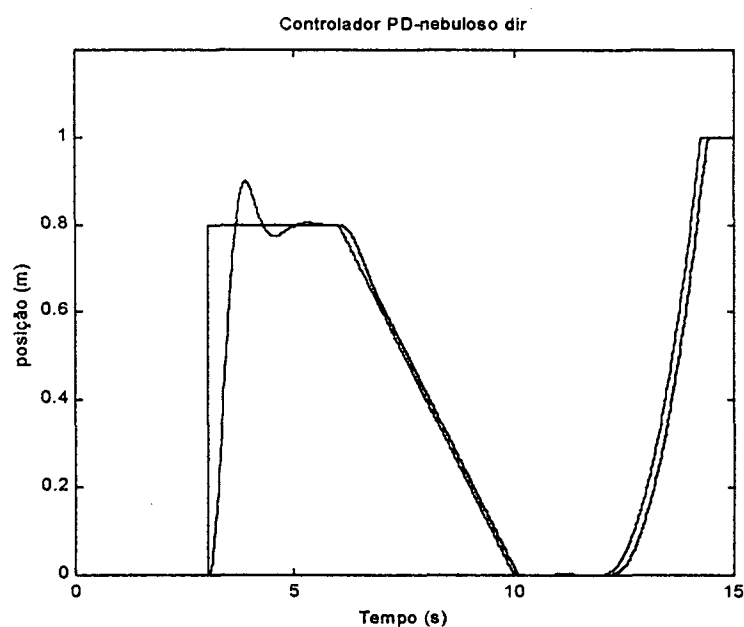


Figura 50 – PID+CLN supervisório direto (Posição)
MASSA: 0,5 kg

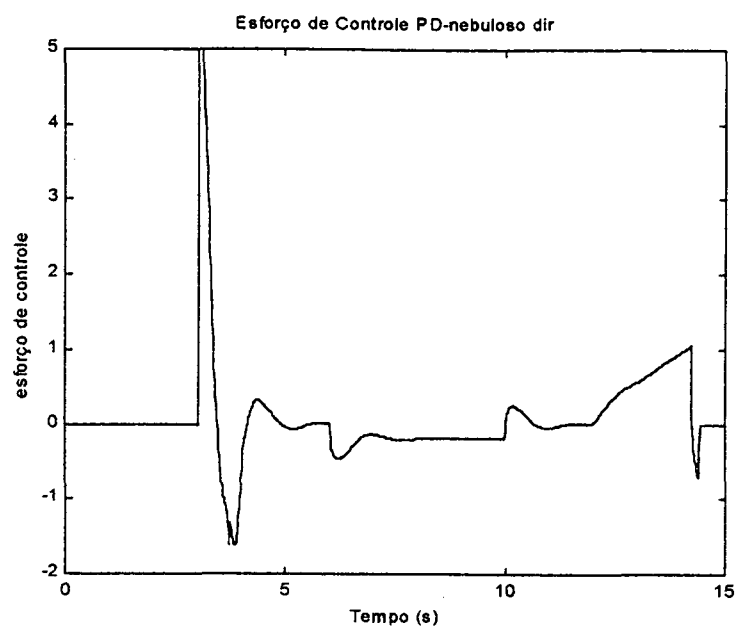


Figura 51 – PID+CLN supervisorio direto (Esforço de controle)
MASSA: 0,5 kg

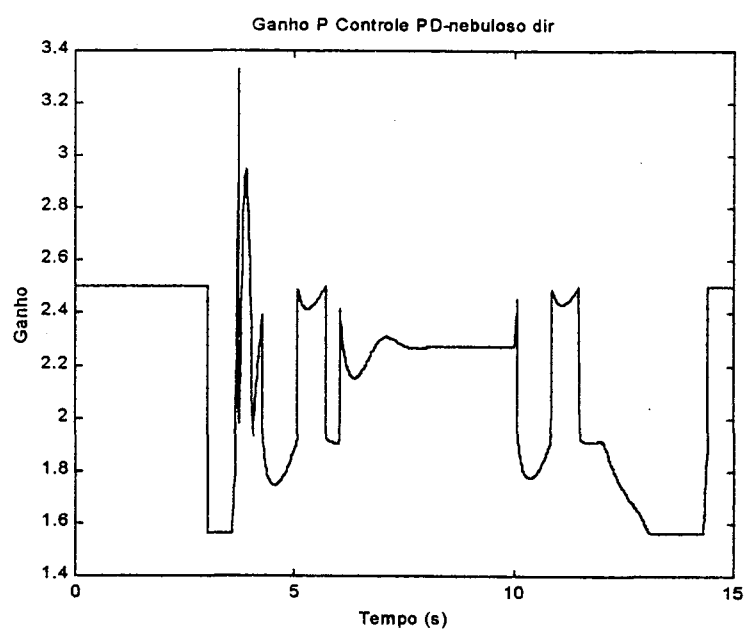


Figura 52 – PID+CLN supervisorio direto (variação do ganho)
MASSA: 0,5 kg

3.4.4 Controlador PID + Nebuloso Supervisório (Inverso)

O controlador PID-CLN supervisório (inverso) possui a mesma estrutura do controlador do item 3.4.3 e mesma base de regras.

Ganhos do Controlador:

- Ganho da Variável *Erro* (GERR): 1,2
- Ganho da Variável *Variação do Erro* (GVERR): 1000
- Ganho da Saída (GSAÍDA): 1,5

Os Ganhos das bandas integral e derivativa foram ajustados usando o método de tentativa e erro (JANTZEN, 1998c) (FRANKLIN et al., 1994):

- $T_D = 0,5$
- $1/T_I = 0,001$

As respostas em malha fechada para este controlador estão apresentadas na seguinte disposição:

	Massa 3,3 kg	Massa 0,5 kg
POSIÇÃO	Figura (52)	Figura(55)
ESFORÇO DE CONTROLE	Figura (53)	Figura (56)
VARIAÇÃO DO GANHO	Figura (54)	Figura (57)

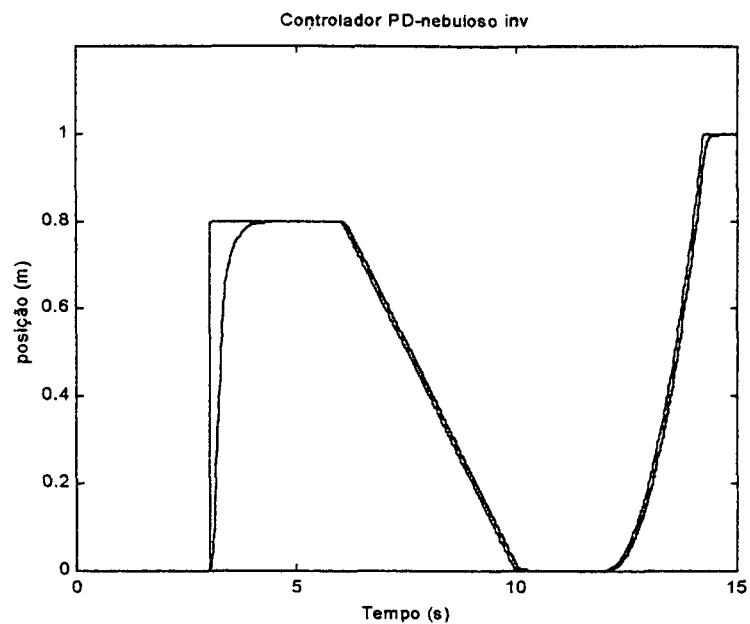


Figura 53 – PID+CLN supervisorío inverso (Posição)
MASSA: 3,3 kg

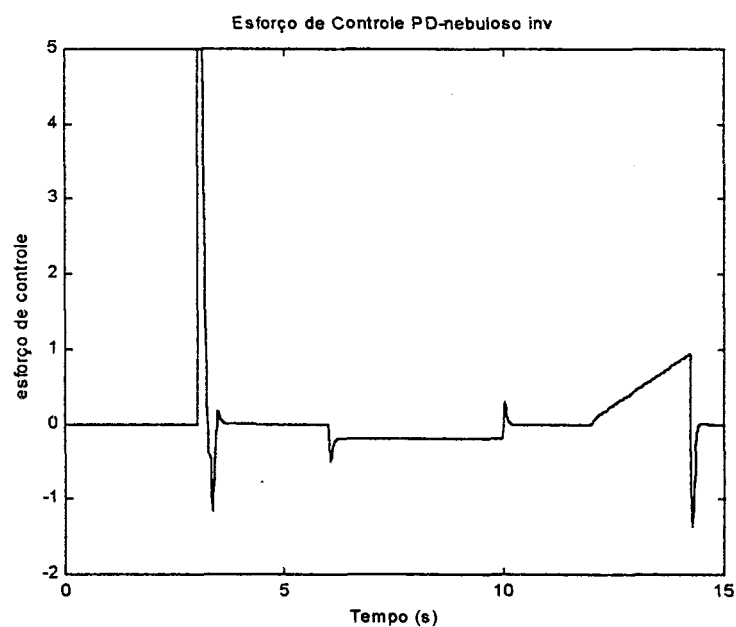


Figura 54 – PID+CLN supervisorío inverso (Esforço de controle)
MASSA: 3,3 kg

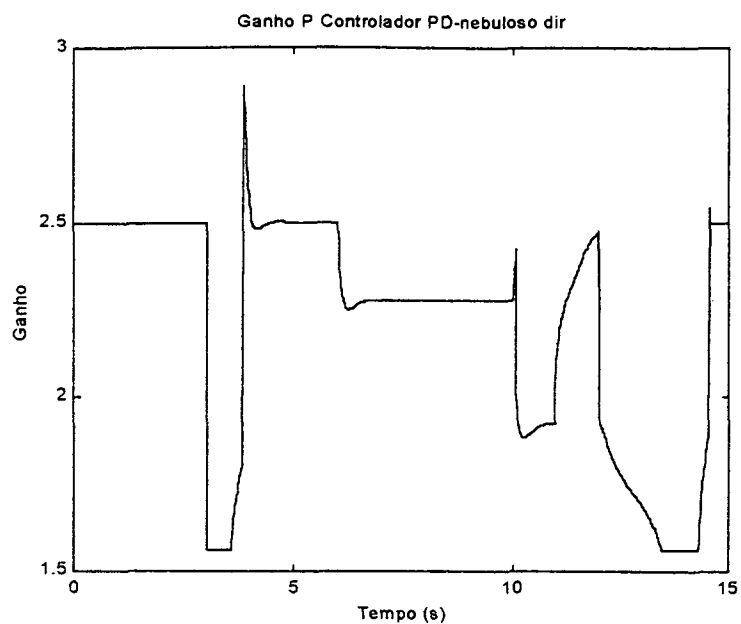


Figura 55 – PID+CLN supervisório inverso (variação do ganho)

MASSA: 3,3 kg

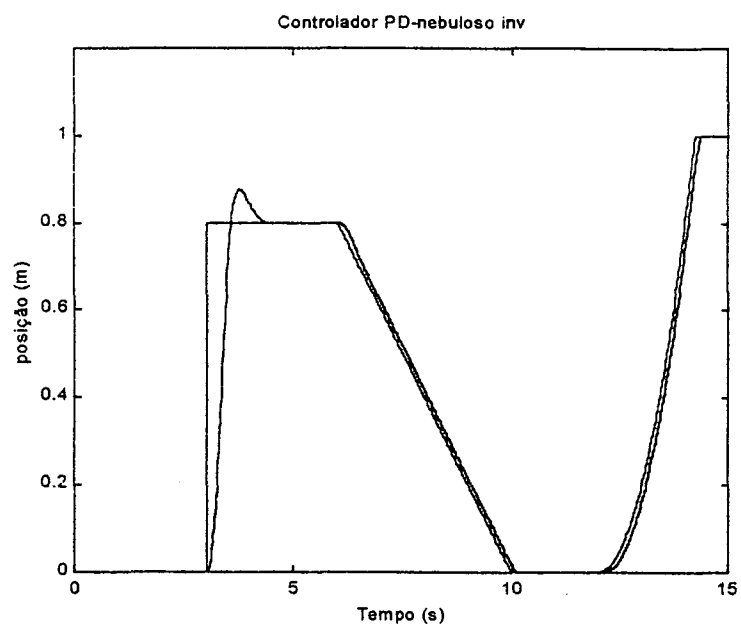


Figura 56 – PID+CLN supervisório inverso (Posição)

MASSA: 0,5 kg

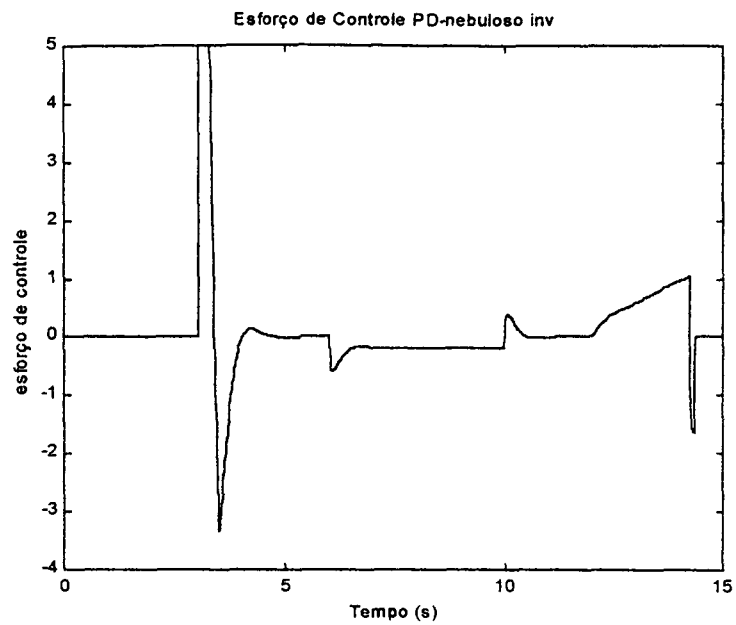


Figura 57 – PID+CLN supervisório inverso (Esforço de controle)
MASSA: 0,5 kg

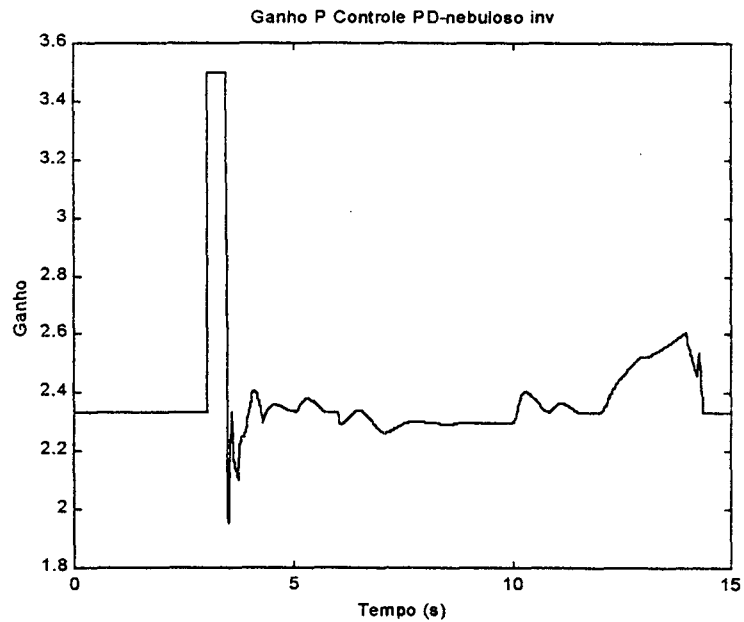


Figura 58 – PID+CLN supervisório inverso (variação do ganho)
MASSA: 0,5 kg

3.4.5 Controlador Nebuloso-PI + PD clássico

Utilizando-se como referência a arquitetura de um controlador CLN-PI + PD clássico em COELHO e COELHO (1999) (ver Figura (58)). Implementou-se um controlador com as mesmas características. Sendo que a realimentação derivativa foi mantida, ao invés da compensação *lead*. Intende-se que para uma aplicação real a compensação *lead* trará os benefícios de diminuição da amplificação em altas frequências, minimizando a sensibilidade a ruídos. (KATZ, 1981 e FRANKLIN et al., 1994)

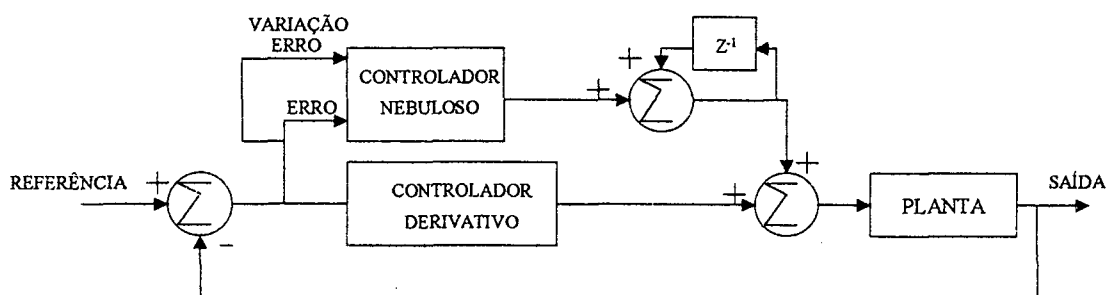


Figura 59 – Arquitetura do CLN-PI + PD clássico

Para a implementação do controlador CLN-PI utilizou-se a mesma estrutura do controlador nebuloso do item 3.4.1.

A base de regras para este controlador encontra-se na Tabela (4).

		var erro				
		ng	nm	ze	pm	pg
e r r o	ng	ng	nm	ng	ng	ng
	nm	pm	nm	nm	nm	ng
	ze	ze	ze	ze	pm	pm
	pm	ng	pm	pm	pm	pg
	pg	pg	pg	pg	pg	pg

Tabela 4 – Base de Regras para CLN-PI .

Ganhos do Controlador:

CLN-PI

- Ganho da Variável *Erro* (GERR): 1,8
- Ganho da Variável *Varição do Erro* (GVERR): 900
- Ganho da Saída (GSAÍDA): 0.092

PD

- Ganho: 0,37

A superfície de controle gerada por este banco de conhecimento pode ser vista na Figura (59).

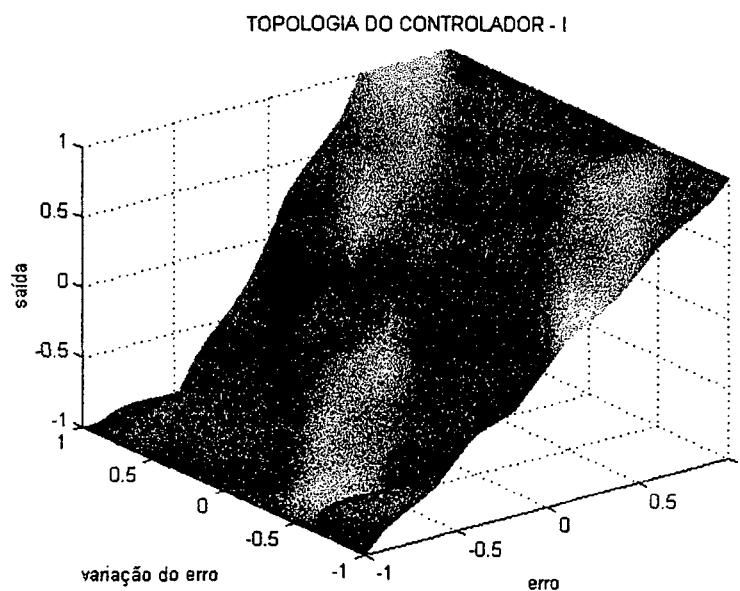


Figura 60 - Topologia do CLN-PI .

As respostas em malha fechada para este controlador estão apresentadas na seguinte disposição:

	Massa 3,3 kg	Massa 0,5 kg
POSIÇÃO	Figura (60)	Figura(62)
ESFORÇO DE CONTROLE	Figura (61)	Figura (63)

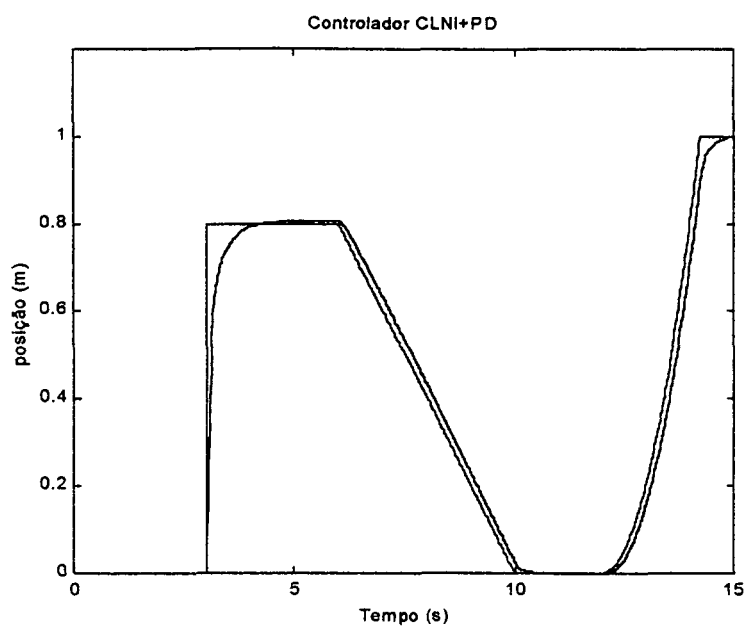


Figura 61 – CLN-PI + PD clássico (Posição)

MASSA: 3,3 kg

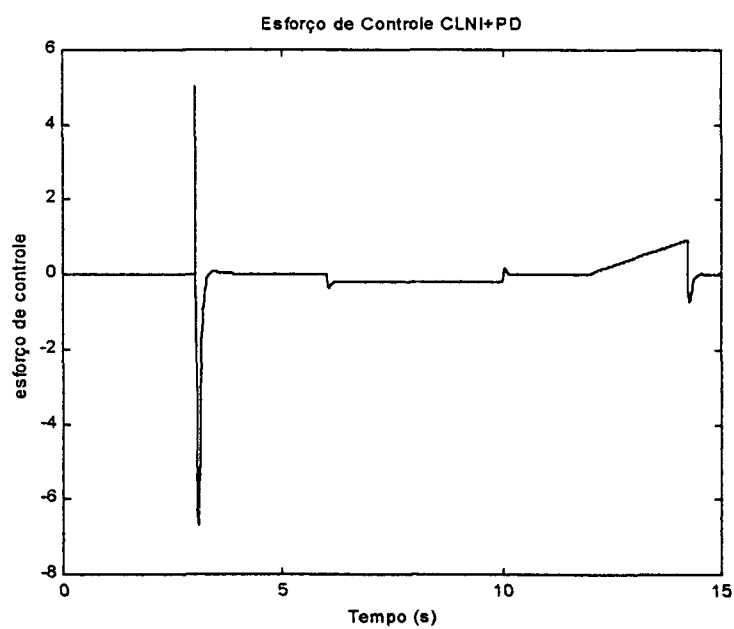


Figura 62 – PID+CLN supervisorio inverso (Esforço de controle)

MASSA: 3,3 kg

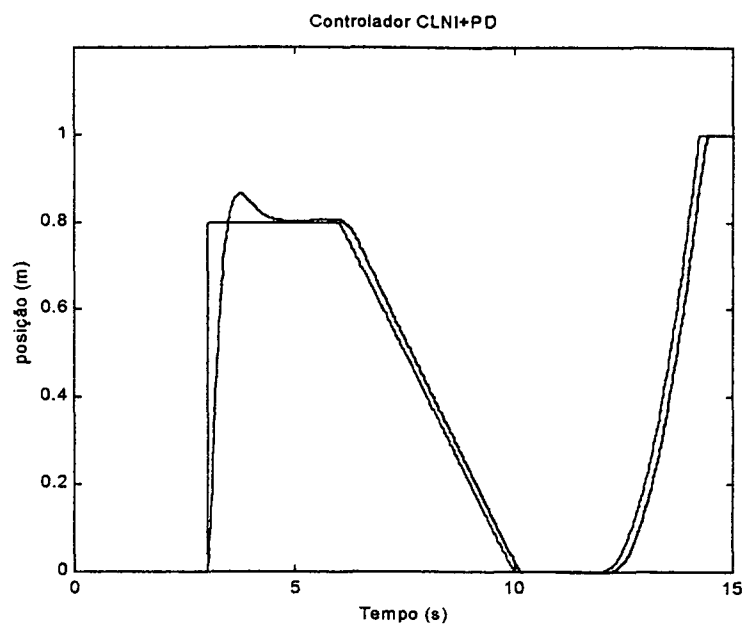


Figura 63 – CLN-PI + PD clássico (Posição)

MASSA: 0,5 kg

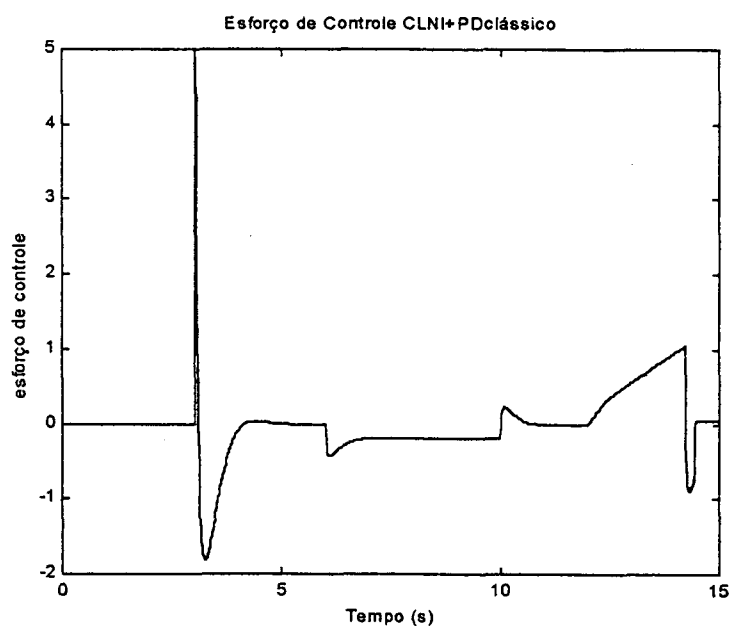


Figura 64 – CLN-PI + PD clássico (Esforço de controle)

MASSA: 0,5 kg

3.5 AVALIAÇÃO DO DESEMPENHO

Os critérios estabelecidos para a avaliação comparativa dos controladores foram listados no item 3.4 . Ou seja:

- *Rise time*
- *Overshot*
- Somatório Absoluto do Erro $(\sum |e| = \sum_{k=1}^n |e(k)|)$
- Variância da ação de controle $(J = \frac{1}{N} \sum_{k=1}^N [u(k) - \bar{u}]^2)$

A partir dos resultados obtidos nas simulações, os parâmetros para avaliação do desempenho dos controladores foram calculados e apresentados na Tabela (6).

	ÍNDICE	CLN	PID clássico	PID+CLN supervisório direto	PID+CLN supervisório inverso	CLN-PI + PD
M =3,3kg	SOMA ERRO	95,23	158,94	142,26	118,81	129,73
	OVERSHOT (%)	5,07	6,3	0,11	—	0,71
	RISE TIME (s)	3,54	3,45	4,03	5,05	4,29
	VARIÂNCIA	0,3506	0,1614	0,2324	0,3305	0,278
M=0,5kg	SOMA ERRO	157,52	262,94	193,94	156,74	166,4
	OVERSHOT (%)	15,44	25	12,54	9,6	8,2
	RISE TIME (s)	3,51	3,54	3,66	3,57	3,48
	VARIÂNCIA	0,5835	0,3537	0,4802	0,7138	0,2322

Tabela 5 – Resultados dos controladores nas simulações.

Analisando os resultados torna-se evidente que todos controladores se comportam de forma satisfatória as perturbações de carga. Sendo que o controlador CLN foi quem apresentou melhor dinâmica, exibindo baixos valores para o somatório do erro, *rise time* e *overshot*. No entanto, o mesmo CLN apresentou a variância do sinal de controle elevada, mas como trata-se da aplicação em um sistema servopneumático, onde a incidência de contato entre as partes móveis é baixo, o valor encontrado pode ser considerado admissível.

O controlador PID clássico apresentou resultados satisfatórios, apenas apresentando um elevado *overshot* na simulação com baixa carga. Este controlador também se destacou pela baixa variância do seu sinal de controle. Um alto valor de *overshot* sugere a possibilidade deste controlador estar sendo afetado pelo fenômeno de *windup* da banda integral do controlador, já que o sistema pneumático possui saturações de sinal como uma de suas características. Observando os resultados deste controlador conclui-se que o *windup* não está ocorrendo, pois o sinal de controle não saturou. Em caso de observação da ocorrência deste fenômeno, principalmente quando o sistema estiver submetido a cargas mais elevadas, arquiteturas *antiwindup* podem ser implementadas. (Franklin et al., 1994)

Obteve-se um resultado importante com o controlador CLN-PI+PD, que apresentou um baixo valor de *overshot* e variância de sinal de controle nas duas simulações. Este resultado sugere uma significativa robustez no controle. No entanto, no parâmetro somatório absoluto do erro, o controlador, apresentou um valor mediano.

Os controladores supervisórios apresentaram resultados satisfatórios, principalmente o controlador PID+CLN supervisório inverso. Este controlador apresentou baixo somatório absoluto de erro e *overshot*, mas em contrapartida uma alta variância do sinal de controle. Estas características sugerem um comportamento superamortecido.

Visando a avaliação da robustez dos controladores implementados submetidos a um sistema com não-linearidades geométricas, foi introduzida uma zona morta ao modelo. Esta zona morta representa a utilização de uma servoválvula de centro não-crítico. De acordo com os dados experimentais de VIEIRA (1998) pode-se obter um valor típico de 15% de zona morta para a servoválvula utilizada no seu experimento. Por apresentarem o melhor desempenho, foram selecionados os controladores PID e CLN-PD para esta simulação e os parâmetros estão apresentados na Tabela (6).

	ÍNDICE	CLN	PID clássico
M = 3,3kg	SOMA ERRO	110,39	235,95
	OVERSHOT (%)	5,18	5,51
	RISE TIME (s)	3,54	3,60
	VARIÂNCIA	0,374	0,28
M=0,5kg	SOMA ERRO	162,43	323,09
	OVERSHOT (%)	15,88	25
	RISE TIME (s)	3,54	3,64
	VARIÂNCIA	0,64	0,50

Tabela 6 – Resultados dos controladores nas simulações.

Observando o desempenho dos controladores PID e CLN com a introdução da zona morta, conclui-se que o controlador PID já necessita um ajuste dos seus parâmetros, mais ainda permanece estável. O CLN por sua vez, manteve o desempenho satisfatório.

Após as análises dos resultados obtidos por estes cinco controladores é possível afirmar que a utilização da lógica nebulosa para soluções de controle em servopneumática é factível. Frente aos bons resultados dos controladores nebulosos pode-se dizer que a sua robustez, pode contribuir de forma significativa para a solução do controle em sistemas de posicionamento servopneumáticos. Outras arquiteturas utilizando controladores nebulosos podem ser implementadas para a solução de problemas específicos. Como exemplo, podem ser utilizadas as variações de controladores PID-nebulosos apresentadas por COELHO e COELHO (1999).

Os controladores nebulosos apresentam as seguintes vantagens sobre os controladores convencionais:

(LI e LAU, 1989, COELHO e COELHO, 1999 e JANTZEN, 1998c)

- Não necessitam de modelos matemáticos do processo.
- Lidam com sistemas não-lineares, alta-ordem, atraso elevado e pouco amortecidos.
- Facilidade em codificar o conhecimento especialista.
- Grandes limites de controlabilidade.

CAPÍTULO 4

CONCLUSÃO

Sob uma ótica objetiva, pode-se dizer que, as metas traçadas para o desenvolvimento deste trabalho foram atingidas, os resultados obtidos através das simulações sugerem a factibilidade da utilização dos controladores baseados em lógica nebulosa como solução de controle para posicionadores servopneumáticos.

A lógica Nebulosa, uma das ferramentas para a representação do conhecimento especialista, desenvolvida por ZADEH (1965), foi utilizada para a implementação de controladores em diferentes arranjos. Estes se mostraram capazes de controlar satisfatoriamente um modelo de sistema servopneumático submetido a variações de sinal de entrada e perturbação de carga. Sua capacidade de atuação não-linear comprovou que se trata de uma ferramenta importante para a solução de problemas de controle em servopneumática.

Neste trabalho obteve-se um algoritmo mais eficiente para o processo de nebulização. Apresentado velocidade de processamento bem superior aos demais algoritmos testados.

Existem muitos campos ainda passíveis de desenvolvimento na implementação de controladores baseados em lógica nebulosa , principalmente no tocante a sua capacidade de aprendizado. Este campo, hoje dominado pelo uso do Algoritmo Genético e do híbrido neuro-nebuloso, pode ser explorado utilizando outras técnicas de busca, que venham a ser encontradas na literatura especializada.

Uma sugestão de prosseguimento para este trabalho é a otimização dos algoritmos. A utilização do processamento paralelo nos diversos processos do controlador baseado em Lógica Nebulosa, mais precisamente a avaliação de regras, deve ser levada em consideração. Visando a sua utilização suportada por sistemas multi-microprocessados ou com sistemas operacionais preemptivos.

Relativo a aplicações pode-se sugerir trabalhos com sistemas nebulosos para tratar incertezas de medição em sensores. Como também a utilização de modelos nebulosos para implementação de controladores do tipo *sliding mode*, ótimo ou adaptativo inteiramente nebulosos. (NAM, 1994 e Coleman et al., 1994)

REFERÊNCIAS BIBLIOGRÁFICAS

ALTROCK, C. . Fuzzy Logic Merges with Bang-Bang, PID. In Tech, Vol. 46, No. 5 , 1999

ANDERSEN, B. W. The Analysis and Design of Pneumatics Systems. John Wiley & Sons. New York, EUA, 1967.

BARCZAK, C. L.; ROSA, L.. Controlador de Lógica Nebulosa para Posicionadores em Máquina Ferramenta. 1ºSimpósio Brasileiro de Automação Inteligente, Rio Claro. 1993.

BARCZAK, C. L.. Controle Digital de Sistemas Dinâmicos – Projeto e Análise. Notas de aulas da Disciplina do curso de Mestrado. GRUCON – Universidade Federal de Santa Catarina, Florianópolis. 1993.

BITTENCOURT, G.. Inteligência Artificial – Ferramentas e Teorias. Editora da Universidade Federal de Santa Catarina, Florianópolis. 1998.

BOLLMANN, Arno. Fundamentos de Servohidráulica. Universidade Federal de Santa Catarina, Florianópolis. 1994.

BONISSONE, P. P. et alii. Industrial Applications of Fuzzy Logic at General Electric. Proc. IEEE, Vol. 83, No. 3, pp. 450-465, 1979.

BRONSTEIN, I. SEMENDIAEV, K. Manual de Matemática para Engenheiros e Estudantes. Editora MIR. Moscou, Russia. 1979.

BYRIEL, J.. Neural-Fuzzy Classification of Cell in Cervical Smers.. Master's Thesis. Technical University of Dnemark. 1999.

CHEN, Chieh-Li ; CHEN, Pey-chung ; CHEN Cha'o-Kuang. A Pneumatic Model-following Control System Using a Fuzzy Adaptive Controler.. AUTOMATICA Professional. New York.,EUA. 1994.

COELHO, A. A. R. , LUIZ, C. C. , BRUCIAPAGLIA A. H. . Controle Adaptativo versus Controle Fuzzy: Um Estudo de Caso em um Processo de Nível. SBA Controle e Automação, Vol. 8, no. 2, pp. 43-51. 1997

COELHO, L. S., COELHO, A. A. R. Fuzzy PID Controllers: Structures, Design, Principles and Application for Nonlinear Practical Process. Advances in Soft Computing - Engineering Design and Manufacturing, Roy,R., Furuhashi, T. Chawdhry, P. K. (eds.), Springer-Verlag, London, pp.147-159. 1999.

COLEMAN C. P.;LYGEROS J.; GODBOLE D. N. Model Based Fuzzy Logic Control. Proceedings of the IEEE Conference on Decision and Control, pp 2633-2638. 1994.

COLEMAN, C. P. ; GODBOLE, D. . **A Comparison of Robustness: Fuzzy Logic, PID and Sliding Mode Control.** Proceedings of the IEEE-FUZZY pp 1654-1659. 1994.

COX, Earl. **The Fuzzy Systems Handbook.** AP Professional. New York.,EUA. 1994.

FLYNN, D. ; HOGG, B. W. ; SWIDENBANK, E. **Expert Adaptive Control Applied to Turbogenerator Systems.** IEEE Proceedings – Control Theory Appl., Vol 144, No. 1, pp. 2-7, 1995.

FRANKLIN, G. F. ; POWELL, D. ; EMANI-NAEINI, A.. **Feedback Control of Dynamic Systems.** Addison-Wesley. EUA. 1994.

GOMIDE. F. A. C. ; GUDWIN, R. R. ; s. **Modelagem, Cotrole, Sistemas e Lógica Fuzzy .** SBA Controle e Automação, Vol. 4, No. 3. 1994.

INTERNATIONAL ORGANIZATION FOR STANDARTIZATION. ISO 6358 – **Pnematic Fluid Power – Components Using Compressible Fluids – Determination of Flowrate Characteristics.** 1989

JANG, Jyh-Shing R. ; SUN, Chuen-Tsai. **Neural-Fuzzy Modeling and Control.** Proceedings of the IEEE, Vol. 83, No. 3 pp.378-406. March 1995

JANTZEN J. . **Design of Fuzzy Controllers.** Technical Report no. 98-H864. Technical University of Denmark. 1998a.

JANTZEN J. . **Self-Organising Fuzzy Controller**. Technical Report no. 98-H869.
Technical University of Denmark. 1998b.

JANTZEN J. . **Tuning of Fuzzy PID Controllers**. Technical Report no. 98-H871.
Technical University of Denmark. 1998c.

JANTZEN J. . **Fuzzy Supervisory Control**. Technical Report no. 98-H875.
Technical University of Denmark. 1998d.

KATZ, P. **Digital Control Using Microprocessors**. Prentice-Hall, USA. 1981.

KICKERT, W. J. M.. **Application of Fuzzy Controller in a Warm Water Plant**.
Automatica. Vol. 12 pp.301-308. 1976.

KOSANOVIC, B., CHAPARRO, L.F., AND SCLABASSI, R.J. **Signal Modeling
with Dynamic Fuzzy Sets** Proc. IEEE Intl. Conf. on Acoustics, Speech
and Signal Processing, ICASSP-96, VOL. 5, PP. 2829-32, 1996.

KOSKO, Bart. **Neural Networks and Fuzzy - A Dynamical Systems Approach
to Machine Intelligence**. Prentice Hall. EUA. 1992.

LARKIN L. I.. **A Fuzzy Logic in Controller for Aircraft Flight Control**. Industrial
Applications of Fuzzy Control. Elsevier Sci. Pub. pp. 87-103.1990.

LEE C. C.. **Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part 1**.
IEEE Transactions on Systems, Man and Cynbernetics. Vol. 20 pp.404-
418.1990.

LEE, John ; SEARS, Francis W.. **Termodinâmica**. Ao Livro Técnico S.A.. Rio de Janeiro. 1962.

LI Y. F. ; LAU C. C.. **Development of Fuzzy Algorithms for Servo Systems**. IEEE Control Systems Magazine. pp. 65-70. 1989.

MANDANI, E. H.. **Application of Fuzzy Algorithms for Control of a Single Dynamic Plant**. Automatica. IEEE. Vol. 121 pp1585-1588, 1974.

MANDANI, E. H. ; ASSILIAM, S. **A Fuzzy Logic Controller for a Dynamic Plant**. International Journal Man and Machine Studing., Vol 7, pp1-13, 1975.

MENDEL, J. M. **Fuzzy Logic Systems for Enginnering: A Tutorial**. Proc. of the IEEE, Vol 83, pp345-377, 1995.

NAM, S-K. , KIM J-S. **Fuzzy Sliding-Mode Control of Gasoline Fuel-Injection System with Oxygen Sensor**. Series C, JSME International Jornal, Vol. 37 , No. 1, pp. 100-106, 1994.

NIEMELA, E. ; ZHAO T. ; VIRVALO T.. **Fuzzy Logic Applied on a Hydraulic Position Servo**. The 3th. Scandinavian International Conference on Fluid Power. Vol 1 pp155-175. 1993

NOBRE, F. S. M. ; PALHARES A. G. B. **Informações Qualitativas e Quantitativas no Projeto de Controladores Nebulosos**. SBA Controle e Automação, Vol. 8, no. 2, pp. 77-92.

SANTOS , Eduardo Alves. Análise Teórico Experimental de Posicionador Pneumático. Universidade Federal de Santa Catarina, Florianópolis, 1996.
Dissertação de mestrado.

SANO, N. ; TAKAHASHI, R.. Modified and Fuzzified General Problem Solver for “Monkey and Banana” Problem. JSME International Jornal. Vol. 37, No. 1. 1994.

SCHILDT, Herbert. C Completo e Total. Macgraw-Hill. São Paulo. 1990.

SHIH, M. ; TSENG, S.. Pneumatic Servo-Cylinder Position Control by PID-Self-Tunning Controller. JSME International Jornal. Vol. 37, No. 3. 1994.

SPLLIMAN, Richard. Genetic Algorithms. Dr. Dobb’s Jornal. Vol. 197 pp. 26-30. 1993.

SMITH, J. M. ; VAN NESS. Introdução à Termodinâmica da Engenharia Química. Editora Guanabara. Rio de Janeiro. 1949.

STRINGER, J. Hidraulic Systems Analysis – Na Introduction. 1ª Ed. . The Macmillan Press ltd. , 1976.

TAKEUCHI I.; FURUHASHI T.. A Description of Dynamic Behavior of Sensory/Motor Systems with Fuzzy Symbolic Dynamic Systems. Proc. of the Int. Symposium on Artificial Life and Robotics, 1999.

TAN J.; CHANG Z.. **Linearity and a Tuning Procedure for Fuzzy Logic Controllers.** ASAE Transactions of The ASAE. Vol. 37 pp.973-979.1994.

VAN WYLEN, Gordon. **Fundamentos da Termodinâmica Clássica.** Addison Wesley. Guanabara. 1990.

VIOT, Greg. **Fuzzy Logic in C.** Dr. Dobbs's Jornal. Vol. 197 pp. 40-49. 1993.

VIRVALO, T. ; KOSKINEN, H. . **Electro-Pneumatic Servo System Design.** Power International. June/1989.

VIRVALO, T. **Design a Pneumatic Position Servo System .** Power International pp. 141-149. December/1989.

ZADEH, L. A.. **Fuzzy Sets.** Information and Control. Vol. 8. pp. 338-353. 1965.

ZADEH, L. A.. **Making Computers Think like People.** IEEE Spectrum, pp 26-32, 1984.

ZUMBRÄGEL, Joaquim. **Flexible Servopneumatische Greifersysteme für die Handhabungstechnik.** Tese de Doutorado. IHP-RWTH. Aachen, Alemanha. 1990.

APENDICE

LISTAGENS

FDS

```

/*Procedure: Fuzzyficacao Direta Simples */
/* Autor: Eng. Marco Antonio Siiva Picanco */
/* Trabalho p/ dissertacao de mestrado 04/95*/

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>

void fuzzy(double e);

double a=4;

double pt_1=-1.0,pt_2=-.5,pt_3=-.75,pt_4=-.25,pt_5=-.5,pt_6=0,pt_7=-.25,pt_8=.25
        ,pt_9=0,pt_10=.5,pt_11=.25,pt_12=.75,pt_13=.5,pt_14=1;

double
delta_1=0,delta_2=0,delta_3=0,delta_4=0,delta_5=0,delta_6=0;

double
delta_7=0,delta_8=0,delta_9=0,delta_10=0,delta_11=0,delta_12=0,
        delta_13=0,delta_14=0;

double ent=0;

double gp_1=0,gp_2=0,gp_3=0,gp_4=0,gp_5=0,gp_6=0,gp_7=0;

void main() {

clrscr();

for (;;) {
printf("entre o valor de entrada : ");
scanf("%lg",&ent);
printf("\n %f",ent);

printf("\n");

if (ent==3)break;

```

```

do{
outport(956,1);

fuzzy(ent);

outport(956,0);

fuzzy(ent);

} while (!kbhit());

printf("\n%f \n%f \n%f \n%f \n%f \n%f \n%f \n%f\n",gp_1,gp_2,gp_3,gp_4,
gp_5,gp_6,gp_7);

}
}
/* inicio da avaliacao */

void fuzzy(double e){

/* valor nebuloso 1 */

delta_1=e-pt_1;
delta_2=pt_2-e;

if ((delta_1 <=0) || (delta_2 <= 0))
    gp_1=0;

    else
    {
        gp_1 = delta_1<delta_2 ? a*delta_1 : a*delta_2 ;
        gp_1 = 1>gp_1 ? gp_1 : 1;
    }

/* valor nebuloso 2 */

delta_3=e-pt_3;
delta_4=pt_4-e;

if ((delta_3 <= 0) || (delta_4 <= 0))
    gp_2 = 0;

    else
    {
        gp_2 = delta_3 < delta_4 ? a*delta_3 : a*delta_4 ;
        gp_2 = 1>gp_2 ? gp_2 : 1;
    }
}

```

```
/* valor nebuloso 3 */
```

```
delta_5=e-pt_5;  
delta_6=pt_6-e;
```

```
if ((delta_5 <=0) || (delta_6<=0))  
    gp_3=0;
```

```
    else
```

```
    {
```

```
        gp_3 = delta_5<delta_6 ? a*delta_5 : a*delta_6 ;  
        gp_3 = 1>gp_3 ? gp_3 : 1 ;
```

```
    }
```

```
/* valor nebuloso 4 */
```

```
delta_7=e-pt_7;  
delta_8=pt_8-e;
```

```
if ((delta_7 <=0) || (delta_8<=0))  
    gp_4=0;
```

```
    else
```

```
    {
```

```
        gp_4 = delta_7<delta_8 ? a*delta_7 : a*delta_8 ;  
        gp_4 = 1>gp_4 ? gp_4 : 1 ;
```

```
    }
```

```
/* valor nebuloso 5 */
```

```
delta_9=e-pt_9;  
delta_10=pt_10-e;
```

```
if ((delta_9 <=0) || (delta_10<=0))  
    gp_5=0;
```

```
    else
```

```
    {
```

```
        gp_5 = delta_9<delta_10 ? a*delta_9 : a*delta_10 ;  
        gp_5 = 1>gp_5 ? gp_5 : 1 ;
```

```
    }
```

```
/* valor nebuloso 6 */
```

```
delta_11=e-pt_11;  
delta_12=pt_12-e;
```

```
if ((delta_11 <=0) || (delta_12<=0))
    gp_6=0;

    else
    {
        gp_6 = delta_11<delta_12 ? a*delta_11 : a*delta_12 ;

        gp_6 = 1>gp_6 ? gp_6 : 1;
    }

/* valor nebuloso 7 */
delta_13=e-pt_13;
delta_14=pt_14-e;

if ((delta_13 <=0) || (delta_14<=0))
    gp_7=0;

    else
    {
        gp_7 = delta_13<delta_14 ? a*delta_13 : a*delta_14 ;
        gp_7 = 1>gp_7 ? gp_7 : 1;
    }

}
```

VARIÁVEIS:

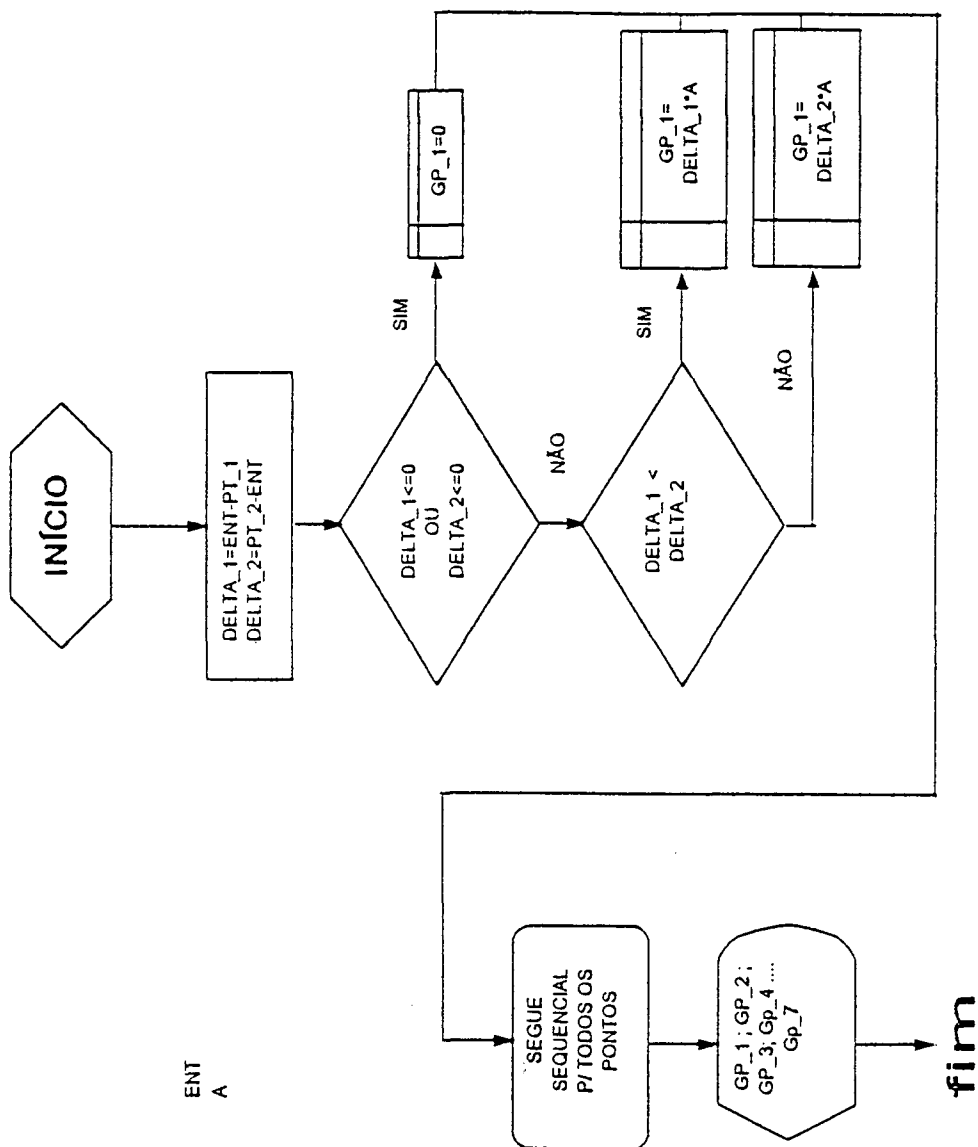
DOUBLE

PT_1
PT_2
PT_3
PT_4
PT_5
PT_6
PT_7
PT_8
PT_9
PT_10
PT_11
PT_12
PT_13
PT_14

DELTA_1
DELTA_2
DELTA_3
DELTA_4
DELTA_5
DELTA_6
DELTA_7
DELTA_8
DELTA_9
DELTA_10
DELTA_11
DELTA_12
DELTA_13
DELTA_14

GP_1
GP_2
GP_3
GP_4
GP_5
GP_6
GP_7

ENT
A



FDSV

```
/* Fuzzificacao Direta Simples com Vetores (FDSV) */
/* Autor: Marco Antonio Silva Picanco */
/* Dissertacao de mestrado 05/95 */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>

struct f_pert {
    double pt_1;
    double pt_2;
    double gp;
};

void fuzzy(double e, struct f_pert *ftp);

void inicial(struct f_pert *ftp);
double a=4, delta_1=0, delta_2=0;

int i;

void main()
{
    int k;
    double ent;

    struct f_pert fp[7];

    clrscr();

    for(;;){
        inicial(fp);

        printf("entre o valor de entrada : ");
        scanf("%lg",&ent);
        printf("\n %f",ent);

        printf("\n");

        if(ent==3)break;

        do{
            outport(956,1);

            fuzzy(ent,fp);

            outport(956,0);

            fuzzy(ent,fp);

        } while (!kbhit());
```

```

for (k=0;k<7;k++)
printf("%f ",fp[k].gp);
}

}

/*funcao de inicializacao*/

void inicial(struct f_pert *ftp)
{
double vet[9]={-1.0,-.75,-.5,-.25,0,.25,.5,.75,1.0};
int n=0;

for(n=0;n<7;n++)
{
ftp[n].pt_1=vet[n] ;
ftp[n].pt_2=vet[n+2];
ftp[n].gp=0;

}

}

/* funcao de fuzzyficacao */

void fuzzy(double e,struct f_pert *ftp)

{
for (i=0;i<7;i++) {

delta_1=e-ftp[i].pt_1;
delta_2=ftp[i].pt_2-e;

if((delta_1<=0)||((delta_2<=0))
ftp[i].gp=0;

else {
ftp[i].gp= delta_1<delta_2 ?(a*delta_1):(a*delta_2);
ftp[i].gp= 1<ftp[i].gp?1:ftp[i].gp;
}

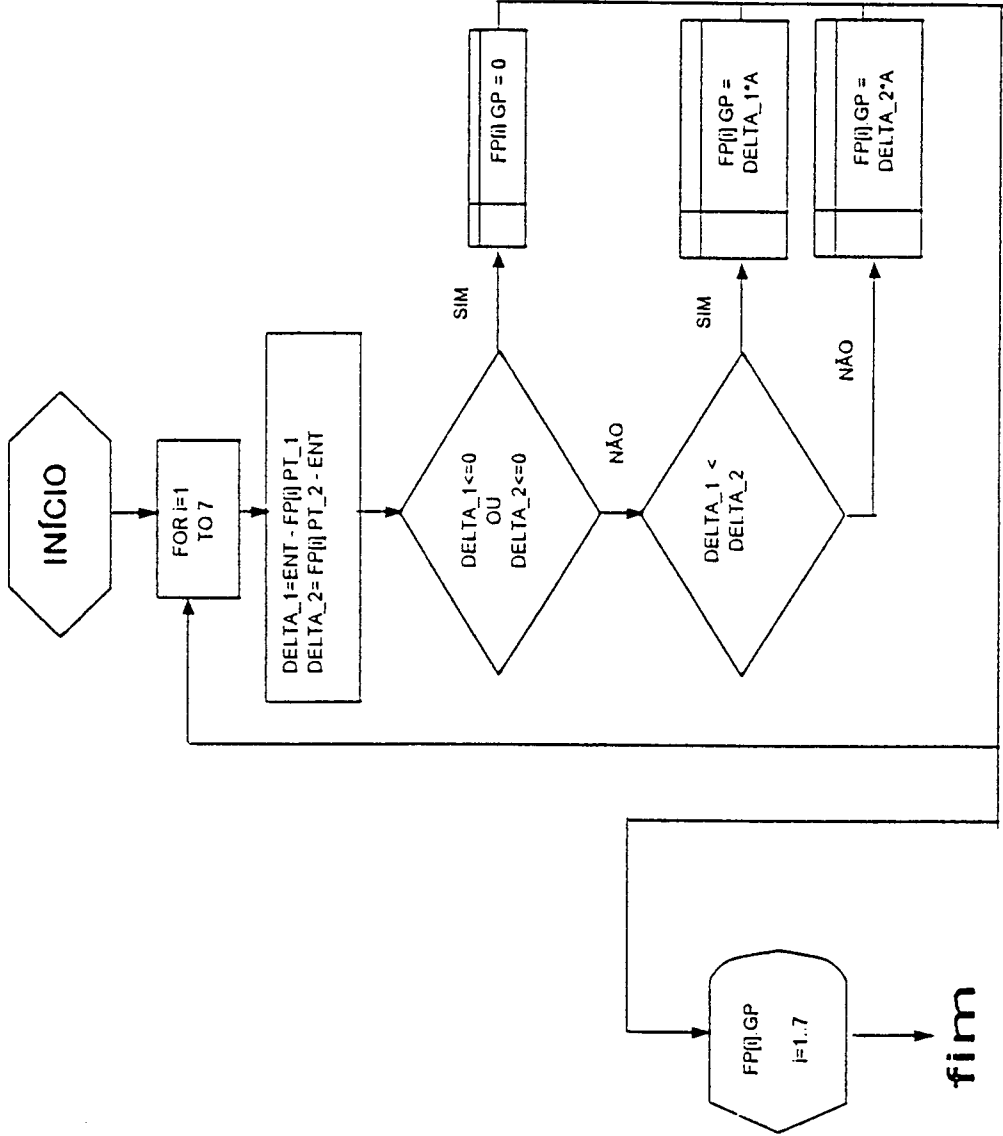
}

}

```

VARIÁVEIS:

DOUBLE
struct
(PT_1 ENT
PT_2 A
GP
) FP[7]



FDL

```
/* Rotina: Fuzzyficacao Direta com Litas (FDL) */
```

```
/* Autor: Marco Antonio Silva Picanco */
```

```
/* Dissertacao de mestrado 05/95 */
```

```
#include <stdio.h>
```

```
#include <dos.h>
```

```
#include <conio.h>
```

```
#include <stdlib.h>
```

```
struct fp_tipo {
    char nome [10];
    float gp;
    float pt_1;
    float pt_2;

    struct fp_tipo *next;
```

```
};
double a=4, ent=0;
```

```
struct fp_tipo *fp;
struct fp_tipo *inicio;
struct fp_tipo *fim;
void fuzzyficacao (float e);
void fnLoad(void);
void fnInicio(void);
```

```
void main () {
```

```
    struct fp_tipo *ftp;
    clrscr();
```

```
    for(;;){
        fnInicio();
        fnLoad();
```

```
        outport(956,0);
        if (ent==3)break;
        do {
            outport(956,1);
            fuzzyficacao(ent);
            outport(956,0);
            fuzzyficacao(ent);
```

```
        } while(!kbhit());
```

```
        for (ftp=inicio;ftp!=NULL;ftp=ftp->next)
            printf("%s -> %f\n",ftp->nome,ftp->gp);
```

```
    }
```

```

}

void fnInicio(void)

{

inico=fim=NULL;

fnLoad();


printf("entre o valor de entrada =>");
scanf("%lg", &ent);
printf("\n %f",ent);
printf("\n");

}


void fnArmazena(struct fp_tipo *ptRegra,struct fp_tipo **fim)

{

if (!*fim) {

                *fim=ptRegra;
                inicio=ptRegra;
            }

else (*fim)->next=ptRegra;

ptRegra->next = NULL;
*fim=ptRegra;
}


void fnLoad(void)

{
struct fp_tipo *info;

FILE *fp;

fp= fopen("fuzzyset","rb");

if(!fp)
{
printf("Arquivo nao pode ser aberto \n");
exit(1);
}

/*libera memoria*/

while(inico){
info = inicio->next;

```

```

free(info);
inicio = info;
}

```

```

inicio=fim=NULL;

```

```

printf("\nloading file\n");

```

```

while(!feof(fp))

```

```

{
info = (struct fp_tipo *) malloc(sizeof(struct fp_tipo));

```

```

if (!info) {
                printf("sem memoria");
                return;
            }

```

```

if (1!=fread(info,sizeof(struct fp_tipo),1,fp))break;

```

```

fnArmazena(info,&fim);
}

```

```

fclose(fp);

```

```

}

```

```

// Rotina para fuzzyficacao

```

```

void fuzzyficacao (float ent)

```

```

{
float delta_1,delta_2;

```

```

struct fp_tipo *ftp;

```

```

for (ftp=inicio ; ftp != NULL ; ftp=ftp->next)

```

```

{
    delta_1=ent-ftp->pt_1;
    delta_2=ftp->pt_2-ent;

    if ((delta_1<=0)||((delta_2<=0))
        ftp->gp=0;

```

```

    else {
        ftp->gp= delta_1<delta_2 ?(a*delta_1):(a*delta_2);
        ftp->gp= 1<ftp->gp?1:ftp->gp;
    }
}

```

```

}

```

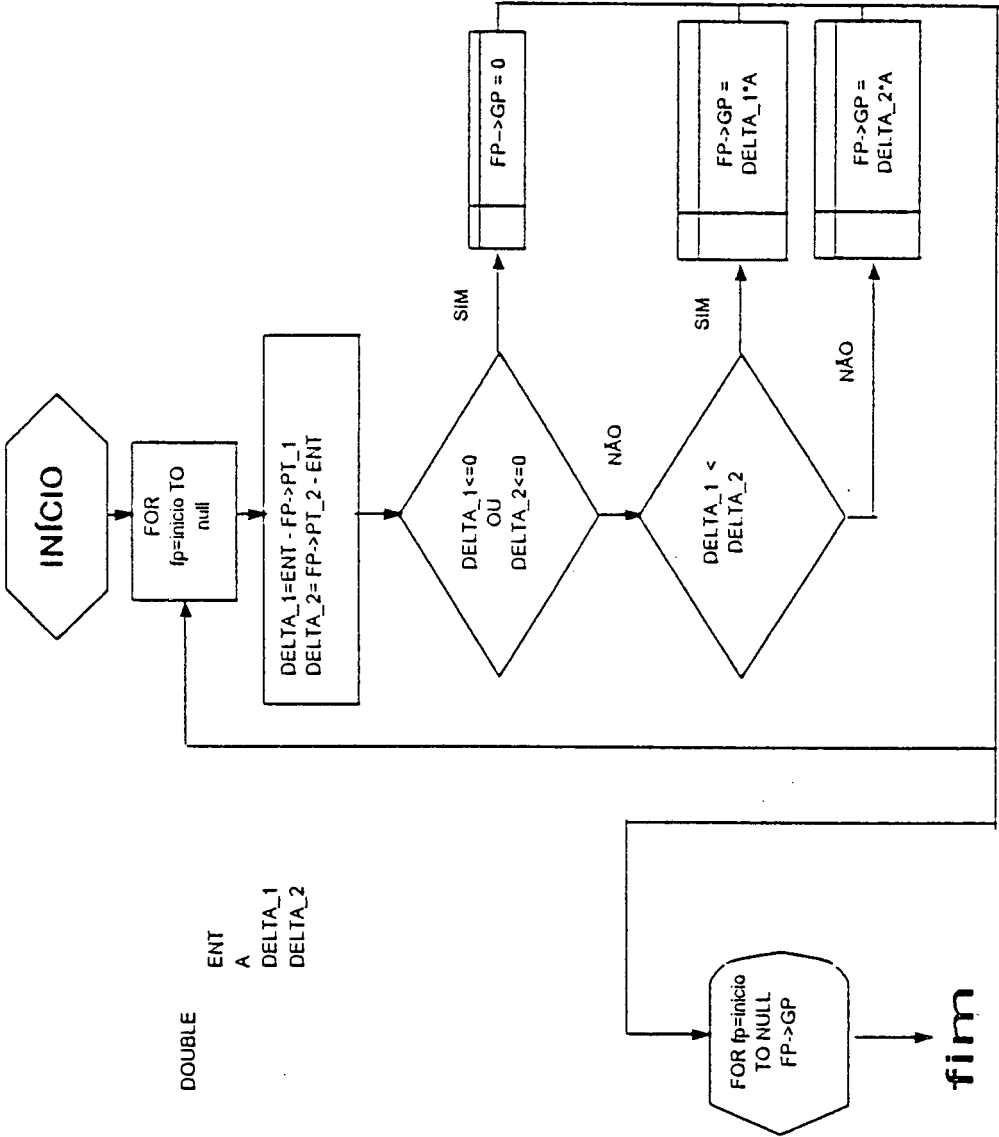
FLUXOGRAMA FDL

FLUXOGRAMA FDL

VARIÁVEIS:

Alocação Dinâmica de Memória.

```
struct fp {
    char nome;
    float GP;
    float PT_1;
    float PT_2;
    struct fp *next;
}
double ENT;
A DELTA_1;
DELTA_2;
```



FMR

```
// FUZZYFICACAO UTILIZANDO META-REGRAS
// AUTOR: MARCO ANTONIO SILVA PICANCO
//TRABALHO PARA DISSERTACAO DE MESTRADO 05/95
```

```
#include <stdio.h>
#include <conio.h>
```

```
double a=4,ent=0;
```

```
double pt_1=-1.0,pt_2=-.75,pt_3=-.5,pt_4=-
.25,pt_5=0,pt_6=.25,pt_7=.5,
      pt_8=.75,pt_9=1;
```

```
double gp_0=0,gp_1=0,gp_2=0,gp_3=0,gp_4=0,gp_5=0,gp_6=0;
```

```
void inicio(void);
void fuzzificacao(double e);
```

```
void main(){
    clrscr();
    for(;;){
        printf("entre o valor da entrada:");
        scanf("%lg",&ent);
        inicio();
        outport(956,0);
        if(ent==3)break;
        do{

            outport(956,1);
            fuzzificacao(ent);
            outport(956,0);
            fuzzificacao(ent);

            } while(!kbhit());

        printf("%f %f %f %f %f %f %f %f ",
              gp_0,gp_1,gp_2,gp_3,gp_4,gp_5,gp_6);

    }
}

void inicio(void)
{

    gp_1=0,gp_2=0,gp_3=0,gp_4=0,gp_5=0,gp_6=0,gp_0=0;
}

void fuzzificacao(double e)
{

    if (e>=pt_5)
    {
```



```

if (e==pt_5) gp_3=1;
else

    if(e>=pt_7)
    {

        if(e==pt_7) gp_5=1;
        else
            if(e>=pt_8)
                gp_6=e>pt_8?(pt_9-e)*a:1;
            else
                {
                    gp_5=(e-pt_7)*a;
                    gp_6=(pt_8-e)*a;
                }
    }
    else
        if(e>=pt_6)
        {
            if(e==pt_6)
                gp_4=1;
            else
                {
                    gp_4=(e-pt_6)*a;
                    gp_5=(pt_7-e)*a;
                }
        }
        else
        {
            gp_3=(e-pt_5)*a;
            gp_4=(pt_6-e)*a;
        }

    }
else
{
    if(e>=pt_3)
    {
        if(e==pt_3)
            gp_1=1;
        else

            if(e>=pt_4)
            {
                if(e==pt_4)
                    gp_2=1;
                else
                    {
                        gp_2=(e-pt_4)*a;
                        gp_3=(pt_5-e)*a;
                    }
            }
            else
            {

```

```
        gp_2=(e-pt_3)*a;
        gp_1=(pt_4-e)*a;
    }

}

else
{
    if(e>=pt_2)
    {
        if(e==pt_2)
            gp_0=1;
        else
        {
            gp_0=(e-pt_2)*a;
            gp_1=(pt_3-e)*a;
        }
    }
    else
        gp_0=(e-pt_1)*a;
}

}

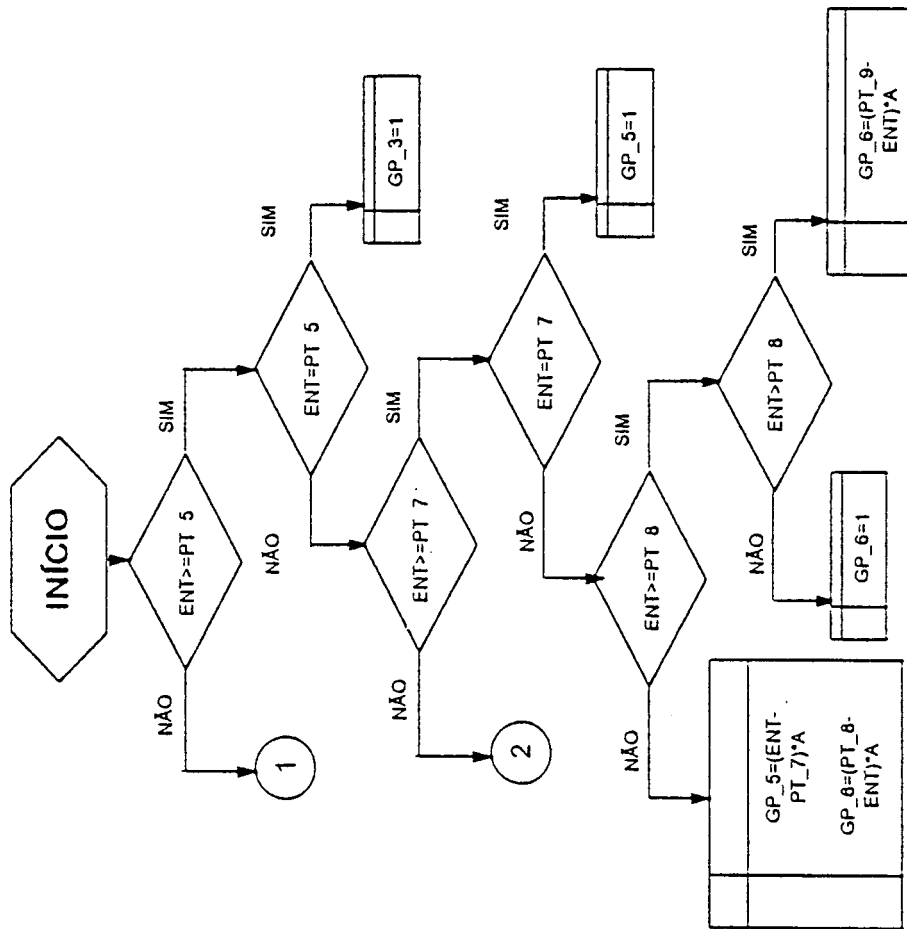
}
```

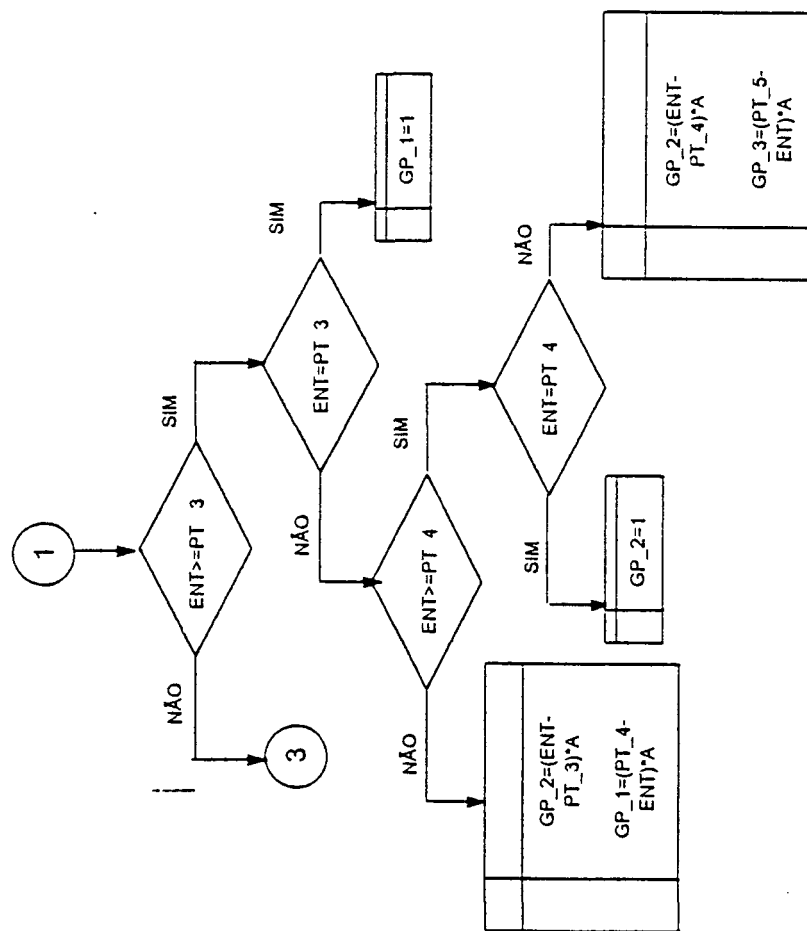
FLUXOGRAMA FMR

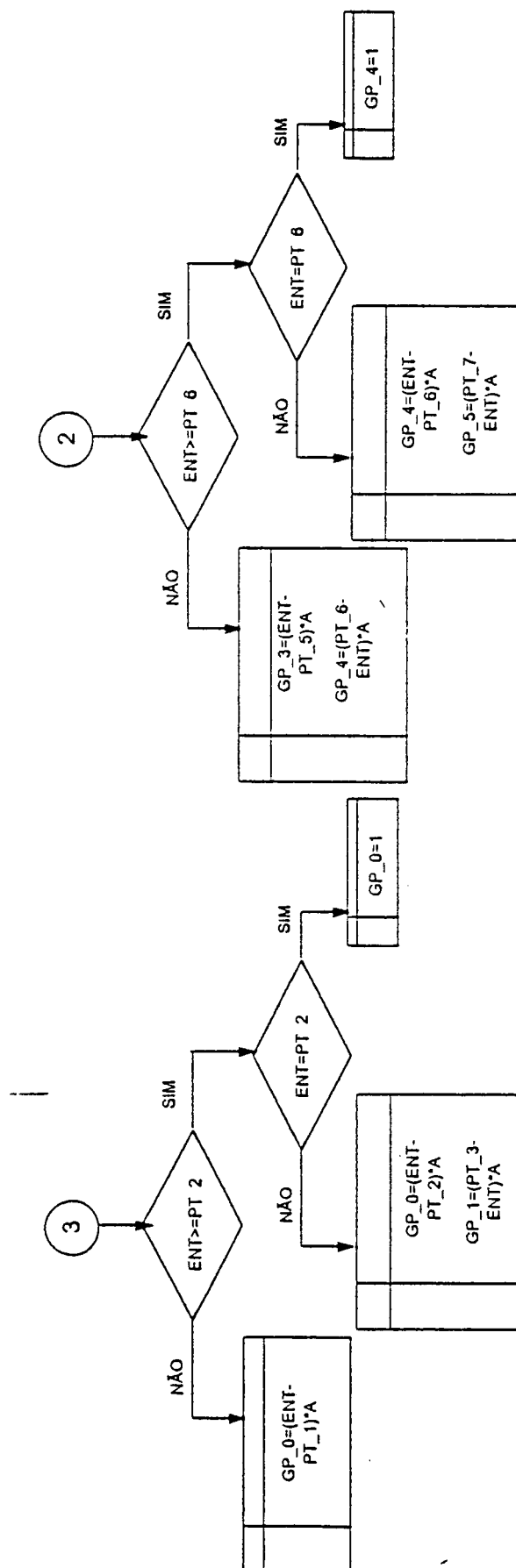
FLUXOGRAMA FMR

VARIÁVEIS:

DOUBLE

ENT
AGP_1
GP_2
GP_3
GP_4
GP_5
GP_6
GP_7PT_1
PT_2
PT_3
PT_4
PT_5
PT_6
PT_7
PT_8
PT_9





FLC

```
// FLC - Controlador baseado em logica fuzzy
//by Marco Antonio Silva Picanço
// Based on Vissim block dll construction.

#include "head.h"
#include <windows.h>
#include <string.h>
#include <math.h>
#include "vsuser.h"
#include "winmem.h"
#include "wprintf.h"
#include "fuzzyfy.h"

#define NNPARAMS 3

// global variables

double D_amplitude=0,D_maxvalue=0,
D_minvalue=0,valorantverr=0,valorant=0,
erro=0,verro=0,saida=0;

char vlnome [5][3];

int i=0,start=0,Flg_inflect=0, Flg_shutdown=0;

far PASCAL _export flc( param, inSig, outSig)
double FAR inSig[], FAR outSig[], FAR param[];
{

//what gets in and what gets out

err.valor=inSig[0]*param[0];
verr.valor=(err.valor-valorant)*param[1];

valorant=err.valor;

valorantverr=verr.valor;

fuzzy(&err);
fuzzy(&verr);

avalr(&err,&verr,&scont);

defuzzy(&scont);

//if (Flg_shutdown==1) scont.valor=0;

outSig[0]=scont.valor*param[2];
```

```

outSig[1]=D_amplitude;

outSig[2]=verr.valor;

outSig[3]=D_minvalue;
outSig[4]=D_maxvalue;

}

/* Called at Sim Start */
void far PASCAL _export flcSS( double FAR param[], long
FAR*runCount )

{

fnLoad();

Flg_shutdown=0;
D_amplitude=0;
valorant=0;
erro=0;
verro=0;
saida=0;
strcpy(vlnome[0],"ng");
strcpy(vlnome[1],"nm");
strcpy(vlnome[2],"ze");
strcpy(vlnome[3],"pm");
strcpy(vlnome[4],"pg");
err.valor=0;
verr.valor=0;
scont.valor=0;

        err.gp[0].centro=-1;
        verr.gp[0].centro=-1;
        scont.gp[0].centro=-1;

        err.gp[1].centro=-.5;
        verr.gp[1].centro=-.5;
        scont.gp[1].centro=-.5;

        err.gp[2].centro=0;
        verr.gp[2].centro=0;
        scont.gp[2].centro=0;

        err.gp[3].centro=.5;
        verr.gp[3].centro=.5;
        scont.gp[3].centro=.5;

        err.gp[4].centro=1;
        verr.gp[4].centro=1;
        scont.gp[4].centro=1;

```

```

for (i=0;i<5;i++)
{
    strcpy(err.gp[i].nome,vlnome[i]);
    strcpy(verr.gp[i].nome,vlnome[i]);
    strcpy(scont.gp[i].nome,vlnome[i]);
    err.gp[i].valor=verr.gp[i].valor=scont.gp[i].valor=0;
}

/* Called at Sim End */
void far PASCAL _export flcSE( double FAR param[], long
FAR*runCount )
{
    ;
}

/* Parameter Alloc
 * paramCount is number of default, VisSim prompted parameters
 * Return total bytes of parameter space for VisSim to alloc,
 * including above params.
 */
long far PASCAL _export flcPA( pCount )
int FAR*pCount;
{
    *pCount = 4; /* prompted param, the rest are unprompted */
    return sizeof(double)*(NNPARAMS+1);
}

/* Parameter Init
 * provide initial values for parameters
 */
void far PASCAL _export flcPI( DOUBLE *param)
{
    param[0]=1;
    param[1]=1;
    param[2]=1;
    param[3]=1;
}

/* Param Set */
char FAR* PASCAL _export flcPC( DOUBLE *param)
{
    return "Ganho da err; ganho da verr; ganho da saida; tolerancia ;";
}

int DLLInst;

/* Required Windows DLL stuff */
int FAR PASCAL LibMain( hInst, dataSeg, heapSize, cmd)
HANDLE hInst; WORD dataSeg, heapSize; char FAR* cmd;
{
    DLLInst = hInst;
    return TRUE;
}

```



```

int FAR PASCAL WEP( parm )
{
    return 1;
}

```

fuzzyfy.c

```

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include "fuzzyfy.h"

```

```

struct gr
{
    char nome[3];
    double valor;
    float centro;
};

```

```

struct variaveis {
    double valor;
    struct gr gp[5];
    float ganho;
};

```

```

struct regras {
    char if1[3];
    char if2[3];
    char then[3];
    struct regras *ptProxima;
};

```

```

struct regras far *inicio;
struct regras far *fim;
struct regras far *ptRegra;
struct regras far *info;

```

// FUZZYIFICAÇÃO BASEADA EM META-REGRAS

```

void fuzzy(struct variaveis *point)

```

```

{

```

```

    struct variaveis *pointer;
    float e;
    int i,a=2;

```

```

pointer=point;

for (i=0;i<5;i++)
pointer->gp[i].valor=0.0;

e=pointer->valor;
if ((e>=-1)&(e<=1))
{
if (e>=pointer->gp[2].centro)
{

if (e==pointer->gp[2].centro) pointer->gp[2].valor=1;
else

if(e>=pointer->gp[3].centro)
{

if(e==pointer->gp[3].centro) pointer->gp[3].valor=1;
else
pointer->gp[3].valor=(pointer->gp[4].centro-e)*a;
pointer->gp[4].valor=(e-pointer->gp[3].centro)*a;
}

else
{
pointer->gp[3].valor=(e-pointer->gp[2].centro)*a;
pointer->gp[2].valor=(pointer->gp[3].centro-e)*a;
}

}
else

if(e>=pointer->gp[1].centro)
{
if(e==pointer->gp[1].centro)
pointer->gp[1].valor=1;
else
{

pointer->gp[2].valor=(e-pointer->gp[1].centro)*a;
pointer->gp[1].valor=(pointer->gp[2].centro-e)*a;

}

}

else
{
pointer->gp[1].valor=(e-pointer->gp[0].centro)*a;
pointer->gp[0].valor=(pointer->gp[1].centro-e)*a;
}

}
else
if(e<=-1)
pointer->gp[0].valor=1;
else
pointer->gp[4].valor=1;
}

```

// AVALIAÇÃO DE REGRAS UTILIZANDO LAÇOS FOR ANINHADOS

```

void avalr (struct variaveis *erro, struct variaveis *verro, struct variaveis *saida)
{
    int i,t,z;
    double peso;

    //Avaliação das Regras (raciocinio fuzzy)
    for (i=0;i<5;i++) saida->gp[i].valor=0;

    for (ptRegra=inicio;ptRegra!=NULL;ptRegra=ptRegra->ptProxima)
    {
        for (i=0;i<5;i++)
        {
            if(!strcmp(ptRegra->if1,erro->gp[i].nome))

                for (t=0; t<5;t++)
                {
                    if
                    (strcmp(ptRegra->if2,verro->gp[t].nome))

                    {

                        peso=min(erro->gp[i].valor,verro->gp[t].valor);
                        if (peso!=0.0)
                        {
                            for (z=0;z<5;z++)
                            {
                                if (!strcmp(ptRegra->then,saida->gp[z].nome))

                                    saida->gp[z].valor=max(saida-
                                    >gp[z].valor,peso);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
}

// Função de desnebulização

void defuzzy(struct variaveis *point)
{
    int i=0;
    double parea=0,sarea=0,area=0;
    for (i=0;i<5;i++)
    {
        if (point->gp[i].valor!=0.0)
        {
            area((((1-point->gp[i].valor)*0.5)*2)+1)*point->gp[i].valor/2;
            sarea+=area;
            parea+= (area*(point->gp[i].centro));
        }
    }

    if (sarea!=0) point->valor=parea/sarea;
}

void fnLoad(void)
{
    FILE *fp;

    fp= fopen("fuzzyset","rb");

    if(!fp)
    {
        printf("Arquivo nao pode ser aberto \n");
        exit(1);
    }

    /*libera memoria*/

    while(inicio){
        info = inicio->ptProxima;
        free(inicio);
        inicio = info;
    }

    inicio=fim=NULL;

    while(!feof(fp))
    {

```

```

info = (struct regras *) malloc(sizeof(struct regras));

if (!info) {
    printf("sem memoria");
    return;
}

if (1!=fread(info,sizeof(struct regras),1,fp))break;

fnArmazena(info,&fim);
}

fclose(fp);

}

void fnArmazena(struct regras *ptRegras,struct regras **fim)
{
if (!*fim) {
    *fim=ptRegras;
    inicio=ptRegras;
}

else (*fim)->ptProxima=ptRegras;

ptRegras->ptProxima = NULL;
*fim=ptRegras;
}

```